UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería en Tecnologías y Servicios de Telecomunicación

TRABAJO FIN DE GRADO

RECONOCIMIENTO BIOMÉTRICO FACIAL BASADO EN NUEVAS ARQUITECTURAS DE APRENDIZAJE PROFUNDO.

Álvar Fernández Arce
Tutor: Aythami Morales Moreno
Ponente: Julián Fierrez Aguilar

Junio 2018

Reconocimiento biométrico facial basado en nuevas arquitecturas de aprendizaje profundo

AUTOR: Álvar Fernández Arce
TUTOR: Aythami Morales Moreno

Biometrics and Data Pattern Analytics – BiDA Lab
Dpto. de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio de 2018

Resumen

La demanda de sistemas de reconocimiento facial está aumentando a gran velocidad, implementándose cada vez sus métodos en un mayor número de aplicaciones e incluso llegando a considerarse normal o imprescindibles dentro de algunas instituciones o ámbitos del mercado. Esto ha impulsado fuertemente el desarrollo en este campo en los últimos años, llegando a alcanzarse resultados muy fiables que han llegado a superar el reconocimiento hecho por los humanos en aquellos escenarios en los que las imágenes analizadas habían sido capturadas en condiciones controladas. De cualquier modo, todavía hay mucho rango de mejora en aquellos casos en los que las imágenes se tomaron en entornos no controlados, en los que se plantean un gran número de desafíos para los algoritmos de detección y reconocimiento. Este caso es especialmente claro en el campo de la videovigilancia en el que además se debe identificar dentro de un gran volumen de datos y la calidad de las imágenes suele ser baja, por lo que se necesita que los algoritmos sean rápidos y robustos.

Este TFG se ha dividido en dos partes. En la primera parte se ha desarrollado un demostrador de tecnologías de reconocimiento facial dotado de una interfaz de usuario sobre la que poder evaluar diferentes algoritmos del estado del arte. El sistema desarrollado explota las características extraídas a partir de redes convolucionales entrenadas a partir de grandes bases de datos de caras.

A partir del demostrador desarrollado se han creado dos aplicaciones: a) Un buscador de similitudes en el que se toma una imagen, se detecta la cara y, tras comparar su información con la de las imágenes de una base de datos, te devuelve una lista ordenada con las identidades de las personas presentes en la base de datos con las que tiene un mayor parecido y en la que no necesariamente tiene porqué estar la identidad buscada; b) un sistema de identificación facial en el que a diferencia del buscador de similitudes, el objetivo es identificar a los usuarios capturados a partir de imágenes suyas etiquetadas.

En la segunda parte del TFG se tratará de mejorar el rendimiento de los algoritmos de reconocimiento para casos de imágenes captadas en entornos no controlados. Para ello, se proponen diferentes hipótesis, todas ellas basadas en la selección de características dependientes del usuario. Mediante la aplicación de estas hipótesis se han conseguido resultados positivos, alcanzando reducción del error de hasta un 31% en el rendimiento en aquellos casos en los que se cuenta con un número muy reducido de imágenes de galería con las cuales comparar.

Palabras clave

Reconocimiento facial, biometría, redes convolucionales

Abstract

The fast increase in demand for facial recognition systems, which are increasingly having their methods implemented in a greater number of applications and are even being considered normal or essential within some institutions or areas of the market, has fostered the development in this field in recent years, reaching very reliable results that have come to exceed the recognition made by humans in those scenarios in which images analyzed had been captured under controlled conditions. In any case, there is still much room for improvement in those cases with unconstrained images, where a large number of challenges are introduced for the detection and recognition algorithms. This case is especially clear in the video surveillance field in which the individual must also be identified from a large volume of data and the quality of the images is usually low, so it is necessary for the algorithms to be fast and robust.

This Bachelor Thesis has been divided in two sections. In the first one, we develop a facial recognition technologies demonstrator consisting of a user interface on which different algorithms from the state of the art will be evaluated. The developed system will work with the features extracted from convolutional networks which were trained with large amounts of faces.

Two applications have been developed to run on the interface: a) a face retrieval which takes an image, detects the face within it and after comparing its information with the information from the data base, it returns an ordered list with the identities from the data base according to the resemblance with the entry image. It looks for the most similar, it does not need to have the same identity. b) a facial identification system in which unlike the face retrieval system, the goal is to identify users captured from labeled images of the user.

The second section of this Bachelor Thesis is centered on the improvement of the facial recognition systems from unconstrained images. Different hypotheses are presented, all of them based on the selection of user-dependent features. By using these hypotheses, a decrease on the error rate of up to 31% has been obtained in those cases in which there is a limited number gallery of images with which to compare.

Keywords

Facial recognition, biometrics, convolutional networks

Agradecimientos

En primer lugar quería agradecerle a mi familia, por estar siempre a mi lado y apoyarme en todas las decisiones que he tomado a lo largo de los años.

También agradecer a mi tutor Aythami, por la ayuda constante que me ha ofrecido durante el desarrollo de este TFG y por darme la oportunidad de embarcarme en este proyecto.

Por último lugar a gradecer a mis amigos, tanto los que estaban antes como los que he conocido a lo largo de la carrera y me han hecho compañía y han sufrido conmigo, sobre todo a Julia por ser mi compañera de laboratorio y fiestas brasileiras durante toda la carrera. Gracias a Pati por meterme y acompañarme en el mundo laboral. Gracias a David e Iván por acompañarme en esas largas horas en el laboratorio en la realización de este TFG y gracias a Ana por la silla.
Hay mucha gente a la que me gustaría agradecer aquí, pero no soy bueno escribiendo, ellos que saben quiénes son y que son importantes para mí.

Gracias a todos.

INDICE DE CONTENIDOS

INDICE DE FIGURAS

GLOSARIO

CNN: Convolutional Neural Networks.

DCNN: Deep Convolutional Neural Networks.

HOG: Histogram of Oriented Gradient

GPU: Graphics Processing Unit

ERR: Equal Error Rate

ROC: Receiver Operating Curve

FAR: False Acceptance Rate

FRR: False Rejection Rate

TAR: True Acceptance Rate

1 Introducción

En este capítulo se incluye una breve introducción al reconocimiento facial y se tratará también, con mayor profundidad el uso de las Redes Neuronales Convolucionales (CNNs) en reconocimiento facial, siendo este el tema principal a lo largo de este Trabajo Fin de Grado. Se incluirán secciones explicativas de las motivaciones y objetivos que se pretenden cubrir con este Trabajo Fin de Grado. Para finalizar, se presentará un breve comentario explicativo de las diferentes secciones en las que está dividido este TFG.

1.1 Reconocimiento facial

El reconocimiento biométrico es un área tecnológica que tiene como propósito discriminar automáticamente entre sujetos de un modo fiable y de acuerdo con alguna aplicación basada en una o más señales obtenidas a partir de rasgos físicos o patrones de conducta, como la cara, la huella, el iris, la voz, la firma, etc. [1]. A estos rasgos personales se les conoce como modalidades biométricas.

Una de estas modalidades biométricas es la cara. El objetivo del reconocimiento facial es extraer la mayor cantidad de información posible de las caras, como puede ser la localización, pose, género, identidad, edad y emoción entre otros factores.

1.2 Uso de DCNNs en el reconocimiento facial

Gracias a los últimos desarrollos en las Redes Neuronales Convolucionales Profundas (DCNNs) se han conseguido mejoras impresionantes en el rendimiento en la detección y el reconocimiento de objetos. Esto ha sido posible gracias a la disponibilidad de grandes cantidades de datos etiquetados y al abaratamiento y mejora constante en la potencia de las unidades de procesamiento gráfico (GPUs). Estos desarrollos en las arquitecturas de aprendizaje profundo también han impulsado mejoras en las capacidades de las máquinas a la hora de conseguir hacer de manera automática tareas de detección facial, estimación de la pose, cálculo de puntos de referencia y reconocimiento facial de imágenes y videos en entornos no controlados.

El reconocimiento facial ha sido ampliamente investigado durante más de dos décadas [2], y desde entonces se han desarrollado numerosos algoritmos de reconocimiento facial que consiguen buenos resultados con imágenes que han sido tomadas en situaciones controladas. El problema se plantea al trabajar con imágenes captadas en condiciones no controladas, en las se deben enfrentar a desafíos como variaciones en poses, iluminación, expresión u oclusión entre otros. Estas posibles variaciones se pueden observar en la ilustración Ilustración 1.1. Esto hace que trabajar con imágenes captadas en condiciones no controladas se traduzca en un empeoramiento importante del rendimiento de los sistemas de reconocimiento.

Uno de los casos más claros de captura de imágenes en condiciones no controladas es la aplicación en videovigilancia, en donde, además, el sujeto tiene que ser identificado de entre

cientos de videos de baja resolución por lo que el algoritmo de reconocimiento debe ser rápido y robusto.

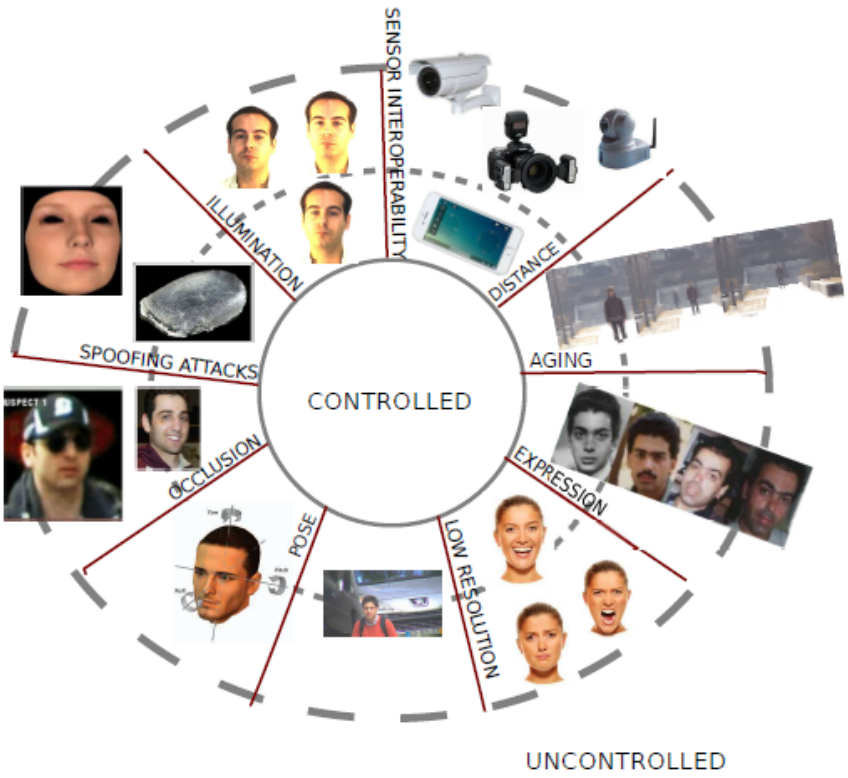

Ilustración 1.1 El reconocimiento facial en imágenes captadas en condiciones no controladas plantea un importante desafío al verse afectado por un gran número de variables. | Fuente: [22]

Para superar estos desafíos, los investigadores han empleado técnicas de aprendizaje profundo para el cálculo de las características requeridas para el análisis facial. En los últimos cinco años, las DCNNs se han usado para resolver numerosos problemas de computer visión como el reconocimiento de objetos [3], [4], [5] y la detección de objetos [6], [7], [8]. Una DCNN típica consiste en una jerarquía de capas convolucionales con una *rectified linear unit activation function*, compuesta de millones de parámetros entrenables. Las DCNNs han conseguido recientemente tener éxito al ser aplicadas a la detección facial [9], [10], [11], localización de puntos de interés [9], [11], [12] y reconocimiento facial [13]. Una de las claves más importantes a la hora de conseguir este extraordinario desempeño es la disponibilidad de grandes cantidades de sets de datos de caras capturadas en entornos no controlados como WIDER FACE [19] para la tarea de detección facial y CASIA-WebFace [14], Mega-Face [15], [16], MS-Celeb-1M [17] y VVG-Face [18] para la tarea de reconocimiento. Esta gran cantidad de datos de entrenamiento implica una significativa variación en pose, iluminación, expresión y oclusión, con lo que conseguimos que las DCNNs sean robustas frente a estas variaciones, gracias a lo que puedan extraer características útiles para estas tareas.

1.2.1 Estructura de los sistemas automáticos de reconocimiento facial

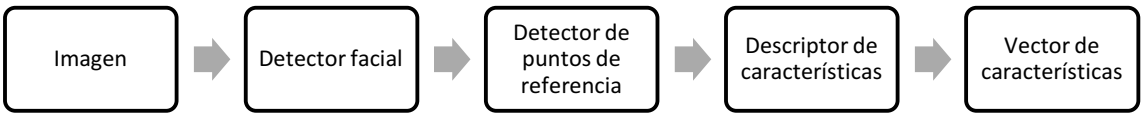

Ilustración 1.2 Estructura de un sistema automático de reconocimiento facial

Normalmente es necesario el uso de tres módulos, como se puede ver en la ilustración Ilustración 1.2 para el funcionamiento de estos sistemas automáticos:

- Primero, se aplica un detector facial para localizar las caras presentes en las imágenes. Para un sistema robusto, el detector facial debería ser capaz de detectar caras con variaciones en las condiciones, como poses, iluminación y escala. Además, los valores de las localizaciones y tamaños de los cuadros que delimitan las caras localizadas son importantes ya que deben estar determinados lo más precisamente posible para que contengan la menor cantidad del fondo de la imagen.
- Segundo, se utiliza un detector de puntos de referencia para la localización de aquellos puntos más importantes de la cara. Con estos puntos se intenta conseguir una normalización de la pose con el objetivo de mitigar los efectos de las rotaciones de planos y escalados. Estos puntos pueden ser el centro de los ojos, la punta de la nariz, límites de la boca, etc.
- Tercero, se utiliza un descriptor de características que codifica la información de la identidad que se extrae de la cara alineada.

Dados los vectores de características obtenidos de las imágenes a comparar, se obtienen puntuaciones de similitud entre ellos mediante alguna de las diferentes medidas de similitud. Si el valor de similitud es menor a un umbral, significa que ambas caras corresponden al mismo sujeto.

Hay dos componentes importantes a la hora de llevar a cabo el reconocimiento facial. El primero es el aprendizaje de características profundas mediante el uso de redes profundas y el segundo es la aplicación de un modelo discriminativo de clasificación en el caso de que el objetivo sea la identificación facial o de una medida de similitud en el caso de que el objetivo sea la verificación facial.

Los métodos de aprendizaje profundo han demostrado posible el aprendizaje de características compactas y discriminatorias mediante el uso de DCNNs entrenadas con grandes sets de datos.

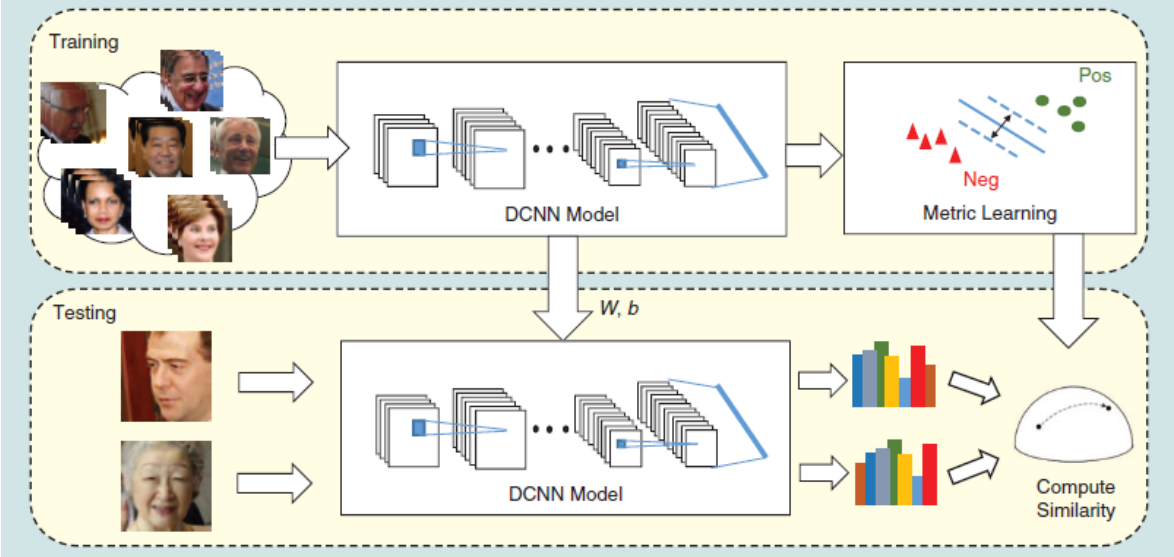

Ilustración 1.3 Un esquema general para el entrenamiento y testeo de un sistema de reconocimiento mediante el uso de DCNNs. | Fuente: [24]

El reconocimiento facial puede dividirse en dos tareas, siguiendo ambas el esquema que se puede observar en la ilustración 1.3: la primera, verificación facial, donde dadas dos imágenes, se tiene como objetivo predecir si las caras presentes en ambas imágenes corresponden al mismo individuo. La segunda, verificación facial, donde dada una imagen de una cara de la que no se conoce la identidad, el objetivo es determinar la identidad del individuo mediante la comparación de sus características con una base de datos. Para ambas tareas, la obtención de características de identidad discriminativas y robustas es crucial.

Para *la verificación facial*, primero se localizan las caras mediante el detector facial y se hace una normalización de la pose de las caras. A continuación, ambas caras se pasan por una DCNN para obtener sus características. Una vez obtenidos los vectores de características de ambas imágenes, se comparan mediante un modelo de similitud como la distancia euclídea o la similitud coseno.

Para *identificación facial*, las imágenes del conjunto de galería se pasan por DCNNs y los vectores de características extraídos de cada una se guardan en una base de datos. Cuando se quiere realizar la *identificación facial* sobre una imagen, se le extraen los vectores de características y se obtiene un valor de similitud con cada una de las identidades de la base de datos.

1.2.2 Detección facial en imágenes en condiciones no controladas

La detección facial tiene un papel muy importante en el esquema de reconocimiento facial y es el primer paso en un sistema automático de reconocimiento facial. Dada una imagen de entrada, un detector facial encuentra todas las caras en una imagen y devuelve, para cada una de ellas, un conjunto de coordenadas que las delimiten. El problema que se encontró en la detección de caras en condiciones no controladas fue que no se podía obtener información de buena calidad frente a las posibles problemáticas presentadas en la ilustración Ilustración 1.1 mediante el uso de características más tradicionales como *Haar Wavelets* o *Histogram of Oriented Gradient* (HOG). Esto se está solucionando gracias a los recientes avances en técnicas de aprendizaje profundo y la mejora en las GPUs, que han hecho posible el uso de DCNNs para la extracción de características. Se ha demostrado [3] que una DCNN

preentrenada con un conjunto grande de datos genéricos puede usarse como extractor de características obteniendo buenos resultados.

1.3 Motivación

En los últimos años, los cambios en una sociedad que tiende a estar cada vez más conectada han hecho que la habilidad de identificar individuos de forma segura, automática y en tiempo real sea prioritario en una gran cantidad de aplicaciones y sea considera un requisito fundamental en numerosas aplicaciones en los ámbitos forense, de videovigilancia o gubernamental entre otros. Esto se ha visto trasladado también a los servicios públicos, donde hay una fuerte demanda, que crece cada día, hacia este tipo de sistemas de reconocimiento, incorporándose en aplicaciones como la realización de pagos usando biometría facial, desbloqueo de sistemas electrónicos, coches inteligentes, etc.

Ha habido grandes avances en el rendimiento de los diferentes algoritmos de reconocimiento facial en los últimos años gracias al uso de las CNNs, pudiendo llegar a asumir que en aquellos casos en los que se cuenta con conjuntos de muestras adquiridas con la ayuda de sujetos en condiciones controladas, la tarea de reconocimiento facial se puede considerar resuelta gracias a los sistemas de biometría del estado del arte actuales. De cualquier modo, estas mejoras y avances en el rendimiento han sido mucho más pronunciadas en el reconocimiento facial en escenarios controlados. En cambio, en escenarios no controlados todavía hay un gran margen de mejora. Una importante línea de investigación en la que hay gran espacio para la mejora es el problema del reconocimiento facial en escenarios de videovigilancia. En estos escenarios las imágenes están tomadas en entornos no controlados, en los que los usuarios no ofrecen ningún tipo de ayuda a la hora de realizar la captura de las imágenes. Por tanto, las variaciones en poses, iluminación, oclusión o diferentes expresiones se encuentran frecuentemente, introduciendo retos reales para los algoritmos de reconocimiento facial.

1.4 Objetivos

En este Trabajo de Final de Grado (TFG) se han querido plantear dos objetivos diferentes:

- Se plantea el desarrollo de un demostrador basado en tecnologías de estado del arte donde integrar los diferentes módulos de los que se compone un sistema de reconocimiento facial. Se hará uso de los métodos más populares del estado del arte. El objetivo es crear una herramienta que permita demostrar las capacidades de la tecnología de reconocimiento facial actual.
- En el contexto del reconocimiento facial en imágenes captadas en entornos no controlados, donde la escasez de datos a partir de los cuales modelar la identidad de un usuario en determinadas aplicaciones reduce el rendimiento de los sistemas de reconocimiento facial, se evaluarán técnicas de selección de características que permitan adaptar las plantillas de los usuarios con el objetivo de mejorar el rendimiento.

1.5 Organización de la memoria

En el capítulo 2 se presenta el diseño de la interfaz visual implementada, así como el conjunto de procesos y algoritmos de los que hace uso para llevar a cabo los diferentes algoritmos de detección y reconocimiento que se han ejecutado sobre ella.

En el capítulo 3 se presentan las diferentes hipótesis desarrolladas con la intención de mejorar los algoritmos de reconocimiento del estado del arte y se exponen los diferentes resultados obtenidos para cada una de estas hipótesis.

En el capítulo 4 se plantean las conclusiones extraídas y se proponen diferentes mejoras en futuras revisiones o ampliaciones sobre el contenido de los capítulos 3 y 4.

2 Demostrador

En este capítulo se explica cuál ha sido el proceso llevado a cabo para la creación de una interfaz de usuario sobre la que poder ejecutar diferentes algoritmos de detección y reconocimiento facial. Se hablará con mayor profundidad sobre la función del demostrador, su diseño y su implementación. En el apartado dedicado al diseño e implementación del demostrador, se comentarán las partes que lo componen, la relación existente entre ellas y su implementación. Se comentará la parte visual e interactiva del demostrador, que se ha denominado front-end; y la parte algorítmica del demostrador, donde se ejecutan los algoritmos de detección y reconocimiento facial, que se ha denominado back-end. Por último, se explican los casos de uso que se han implementado sobre dicha interfaz, así como sus diferencias en funcionalidad y a la hora de ser implementados. Estos casos de uso son los diferentes algoritmos de detección y reconocimiento implementados en el back-end de la aplicación.

2.1 Objetivo del demostrador

La función del demostrador es proveer de una interfaz de usuario sobre la que poder demostrar las capacidades de la tecnología de reconocimiento facial actual. Esta interfaz permite ejecutar los diferentes algoritmos de detección y reconocimiento facial de un modo más simple, visual y accesible para el usuario. De este modo, el usuario no necesita conocer el funcionamiento o la implementación de los algoritmos ejecutados, sino que puede ejecutar y ver los resultados de los algoritmos de reconocimiento directamente.

2.2 Diseño del demostrador

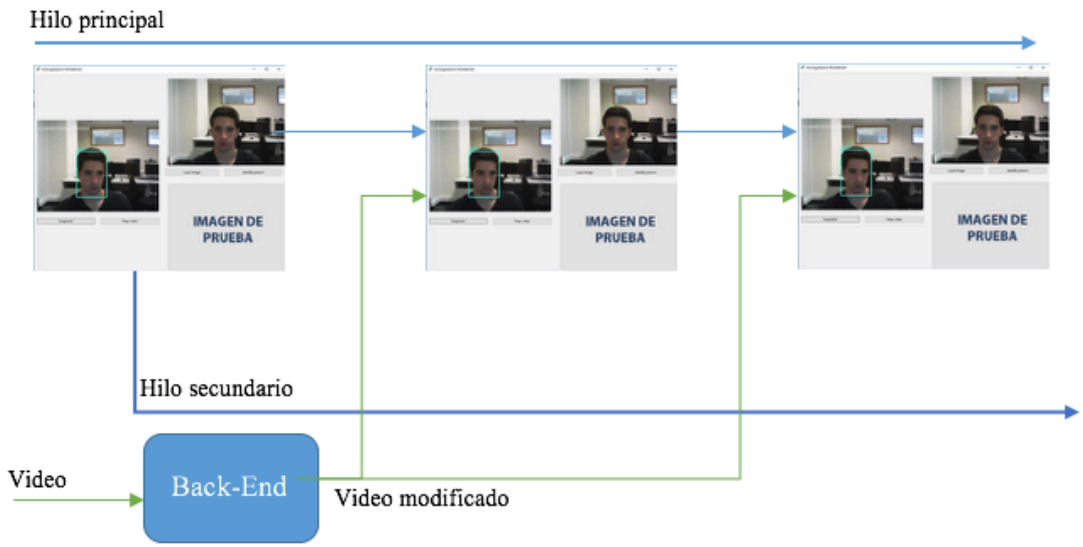

Ilustración 2.1 Representación del diseño del demostrador en el que se pueden observar los dos módulos que lo forman, front-end y back-end, así como las relaciones entre ellos.

El diseño de este demostrador está formado por la unión de dos módulos. Estos módulos se encuentran claramente diferenciados entre sí, pero existe un intercambio de datos entre ellos. Por una parte, se tiene el front-end del demostrador, en donde se puede encontrar todo aquello relacionado con la toma de datos, representación de datos e interacción con el usuario. Por otra parte, se encuentra el back-end del demostrador, en donde se pueden encontrar el conjunto de algoritmos que hacen posible la detección y el reconocimiento facial. El esquema que forman y las relaciones que hay entre ellos están representadas en la ilustración Ilustración 2.1.

2.2.1 Front-end

En el front-end del demostrador es se puede encontrar todo aquello con lo que tiene contacto directo el usuario del demostrador, por tanto, aquello relacionado con la interacción con el usuario y la representación de los resultados visuales. Esto comprende todo lo relacionado con la interfaz creada, con las diferentes secciones y elementos que lo forman, así como las diferentes funcionalidades asignadas a los elementos con los que puede interactuar el usuario, como pueden ser los botones o los paneles entre otros. También comprende la representación visual final con los resultados obtenidos gracias a los algoritmos de detección y reconocimiento aplicados sobre cada uno de los frames del video.

2.2.1.1 Implementación del front-end de la interfaz

La interfaz, al igual que todo el código en este TFG, ha sido desarrollada mediante el uso del lenguaje de programación Python. Además, en el caso de la creación de la interfaz se ha utilizado un paquete de creación de interfaces gráficas de usuario (GUI) llamado Tkinter para la creación de la parte visual e interactiva. Toda la parte del front-end del demostrador ha sido implementada usando un enfoque de programación orientada a objetos debido a exigencias del diseño.

El funcionamiento de la interfaz sigue el siguiente proceso:
Como ya se ha comentado, se ha usado programación orientada a objetos. El total de la interfaz es una clase, que se ejecuta sobre el hilo principal del programa. Sobre este hilo se ejecuta todo lo relacionado con la parte visual e interactiva de la interfaz. A su vez cada una de las secciones de la interfaz, que se explicarán en el apartado 2.2.1.2 Diseño de la interfaz de usuario, son clases en si mismas, de este modo se facilita el borrado o implementación de secciones o funciones en caso de desearlo en futuras implementaciones.

Desde la interfaz, desde el propio hilo principal, se lanza un nuevo hilo secundario, este hilo es sobre el que se ejecutan la captura de video y el back-end de la aplicación además de servir de punto de unión entre back-end y front-end. Es necesario hacerlo de este modo ya que la creación de interfaces mediante Tkinter obliga a crear un hilo secundario en caso de querer ejecutar capturas de video y trabajar con dichas capturas desde la propia interfaz.

El acercamiento que se trató de implementar en un comienzo fue tratar de lanzar otro hilo secundario que corriera paralelamente a la interfaz en el que se cargaría toda la información de la red convolucional. De este modo se podría cargar la información de la red al iniciar el demostrador, hacer que la interfaz no se cargara completamente hasta que no se hubiera

cargado toda la información de la red, y lo más importante, cargar la información de la red solamente una vez en toda la ejecución del demostrador. Esto no fue posible debido a que se producen problemas de ejecución al pasar información de la red entre scripts, lo que obliga a realizar la carga de la red en el mismo script en el que se van a ejecutar los algoritmos de reconocimiento, en este caso, en el único hilo secundario mencionado anteriormente.

El hecho de tener la carga de la red en el hilo secundario, mismo hilo en el que se produce la captura del video, hace que la carga de la interfaz sea lenta, debiéndose cargar por completo la red antes de poderse representar el primer frame de video en la interfaz (frame de video modificado en el back-end). Durante el espacio de tiempo en el que se carga la información de la red, el panel en el que se muestra el video en la interfaz aparece vacío.

En el acercamiento llevado a cabo finalmente, se carga solamente una vez la red, se carga dentro del hilo secundario. El proceso de carga de la información de la red es demasiado lento y costoso como para poder realizarlo en cada nueva ejecución del reconocedor para cada frame. Tampoco era posible pasar los datos desde el script en el que se cargan debido al problema al pasar la información de la red que se ha comentado anteriormente. Al no poder encontrarse una mejor solución a este problema, se perdió la implementación de un botón y un panel de la interfaz como se explicará más adelante.

La parte algorítmica del demostrador, el back-end, está aislado del front-end de tal modo que se pueda cambiar el caso de uso o añadir alguno nuevo sin tener que modificar en nada el front-end de la interfaz. Tan solo serían necesarias unas pequeñas modificaciones en la implementación del caso de uso para su implementación.

2.2.1.2 Diseño de la interfaz de usuario

En este apartado se explica el diseño y estructura visual de la interfaz, aquella parte con la que el usuario interacciona eligiendo las opciones a realizar y en la que ve los resultados visuales. La interfaz está compuesta por un conjunto de paneles y botones, que se pueden observar en la ilustración 2.2. A continuación, se explican detalladamente cada una de las secciones y elementos que lo forman.

Sección izquierda de la interfaz, formada por:

- Panel izquierdo:
  Sobre él se representan los frames de video. Las imágenes representadas en este panel no corresponden directamente al video capturado desde la webcam. El video se captura en el back-end, donde tras ejecutarse los algoritmos de reconocimiento facial, se modifican los frames originales añadiendo la información visual de la localización de las caras detectadas, con un cuadrado; y la identidad asignada a dichas caras, con una etiqueta. Estos nuevos frames modificados son los que se devuelven al front-end del demostrador y se representan sobre dicho panel. Por tanto, este panel de representación es el punto de unión entre el front-end y el back-end del demostrador.
- Botón "Snapshot!":
  Tiene dos funciones. La primera es la captura del frame actual del video y su guardado en un directorio junto con la información del momento en el que se ha realizado la captura. La segunda es el envío de este mismo frame al panel superior derecho de la interfaz para su representación sobre dicho panel.

- Botón “Stop video”:
  Tiene la función de parar la ejecución del hilo secundario. Al pararse el hilo secundario, se interrumpe la captura de video, la representación sobre el panel izquierdo y la ejecución del caso de uso (los casos de uso serán explicados más adelante en el apartado 2.3 Casos de uso).

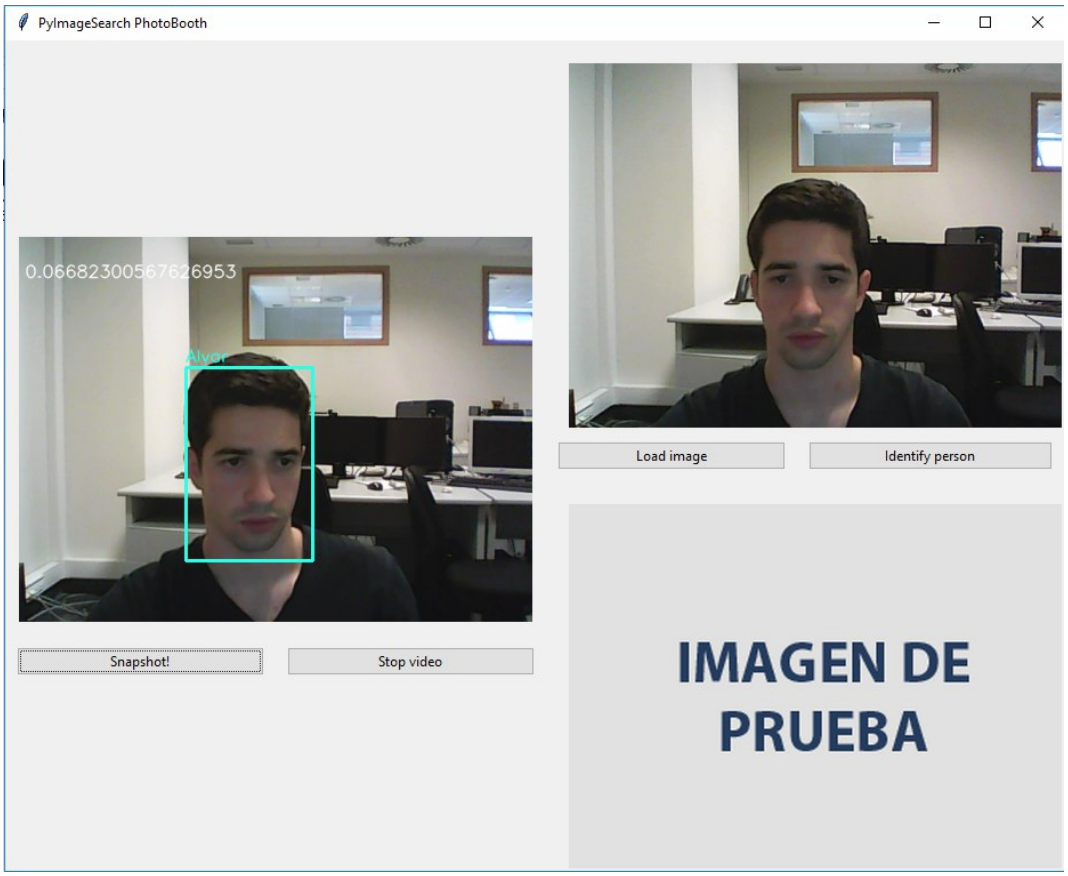

Ilustración 2.2 Interfaz de usuario del demostrador

Sección superior derecha, formada por:

- Botón “Load Image”:
  Este botón permite la carga de una imagen desde un directorio.
- Panel superior derecho:
  Sobre este panel se puede escoger una imagen a representar, hay dos opciones a la hora de escogerla, ambas a partir de implementaciones asignadas a botones cuya funcionalidad ya ha sido explicada. La primera es mediante el botón “Snapshot!” y la segunda opción mediante el botón “Load Image”.

Sección inferior derecha, formada por:

- Botón “Identify person” y Panel inferior derecho:
  Ambos elementos tenían la función de trabajar juntos con el objetivo de tomar la imagen representada en el panel superior derecho, enviársela al back-end del demostrador y ejecutar el algoritmo de reconocimiento sobre la imagen. En este caso, este algoritmo de reconocimiento devolvería una imagen de la base de datos, la imagen con mayor parecido a la presente en el panel superior derecho. Además, el back-end solo debía tomar una cara de las presentes en la imagen del panel superior

derecho, aquella que considerara más importante. Para elegir la más importante tendría en cuenta la información del tamaño y localización de las caras dentro de la imagen.
Finalmente esta implementación no fue posible. La idea original era cargar la red convolucional en otro hilo secundario que corriera en paralelo al hilo sobre el que se ejecuta la interfaz y al hilo sobre el que se captura el video de la webcam y se ejecutan los algoritmos de reconocimiento. El problema que se encontró es que los datos de la red no pueden ser pasados entre diferentes scripts sin producirse el congelamiento del demostrador. Para que funcionaran las implementaciones presentes en el hilo sobre el que corre el back-end del demostrador, se pasó la carga de la red al back-end. Se podría haber hecho esto mismo para la implementación de estos elementos, pero volver a cargar la red es un proceso demasiado lento y costoso que además estropea la usabilidad a nivel de usuario del demostrador.

2.2.2 Back-end

En esta parte es donde se lleva a cabo la ejecución de los diferentes algoritmos que hacen posible el reconocimiento facial. Estos algoritmos de reconocimiento hacen así mismo llamadas a otros algoritmos que se ejecutan por debajo de ellos, como pueden ser los algoritmos de detección facial, extracción de características, comparación de características, etc.

2.2.2.1 Implementación del back-end de la interfaz

El back-end también se encuentra implementado en Python, pero en este caso no se hace uso de Tkinter al no implementarse nada de la interfaz de usuario. Además, la programación deja de estar basada en un paradigma de programación orientada a objetos y pasa a ser funcional. Todos aquellos procesos relacionados con el back-end del demostrador se ejecutan sobre el hilo secundario lanzado desde el hilo principal del demostrador, donde se encuentra la interfaz de usuario.

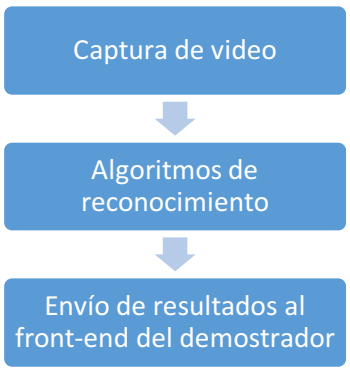

Ilustración 2.3. Esquema de los módulos que forman el back-end del demostrador

El back-end del demostrador está formado por los tres módulos presentes en la ilustración 2.3:

- El primero realiza la captura de video desde una webcam y el envío de los frames pertenecientes a dicho video uno a uno a los algoritmos de reconocimiento.
- El segundo implementa los diferentes algoritmos de detección y reconocimiento. Estos algoritmos toman una imagen a su entrada y devuelven la información de las caras detectadas en ella. Esta información se incrusta visualmente mediante la modificación de las imágenes originales. En el apartado 2.3 Casos de uso se explican las diferentes implementaciones de estos algoritmos de reconocimiento, en concreto los dos casos de uso con los que se ha trabajado: buscador de similitudes e identificador facial.
- La tercera consiste en la toma de las nuevas imágenes modificadas que los algoritmos de reconocimiento devuelven, la unión de todas ellas formando una nueva secuencia de video y el envío de esta secuencia de video al front-end del demostrador para su representación sobre el panel izquierdo de la interfaz de usuario.

2.3 Casos de uso

En este apartado se explican los diferentes casos de uso que se han desarrollado. Estos casos de uso son los algoritmos de reconocimiento que se encuentran en el segundo módulo explicado en el apartado 2.2.2.1 Implementación del back-end de la interfaz. Cada uno de los casos de uso cuenta con una funcionalidad y una implementación diferente.

A la entrada de los casos de uso se tienen los frames del video capturados y pasados desde el primer módulo del back-end del demostrador, pero los algoritmos implementados en los diferentes casos de uso no saben que se tratan de los frames de un video, sino que trabaja con ellos como si fueran imágenes individuales.

2.3.1 Algoritmos de los que parten todos los casos de uso

Los módulos fundamentales de los algoritmos de reconocimiento de los diferentes casos de uso son la detección de caras, extracción de características y comparación de características. Estos módulos serán referidos a lo largo del TFG, por lo que siempre que se hable sobre implementaciones que usen detección de caras, extracción de características o comparación de características, se estará hablando de los módulos aquí presentados.

Detección de caras

Se usará un detector facial propuesto en [20]. Este detector facial integra la detección y el alineamiento facial mediante el uso de CNNs en cascada con aprendizaje multitarea. Primero, produce ventanas candidatas mediante una CNN, después, mediante el uso de otra CNN desecha gran cantidad de estas ventanas y finalmente usa una CNN más poderosa para refinar los resultados.

Para su integración en el demostrador desarrollado, se ha utilizado el código disponible en [23].

Extracción de características

En este caso se usa el modelo pre-entrenado VGG-Face [18] para la extracción de las características. VGG-Face es una red convolucional compuesta por 37 capas, de las cuales 16 son convolucionales, y más de 60 millones de parámetros. Es una red entrenada para reconocimiento facial a partir del modelo VGG (red convolucional generalista de reconocimiento de imágenes). Se pasan las imágenes por la red, sin fine-tunning para la extracción de características de la capa fc6, descartando los valores de la capa fc7 y fc8, obteniendo un vector de características de longitud 4096. En la ilustración 2.4 se puede observar la configuración de la CNN.

En este caso, además, una vez obtenidos los vectores de VGG-Face, se trabajará y se obtendrán resultados con estos vectores de características normalizados y sin normalizar. Se aplicará una normalización basada en el momento estándar.
Para la integración de la red convolucional VGG-Face se ha utilizado el código disponible en [23].

layer	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
type	input	conv	relu	conv	relu	mpool	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	relu	mpool	conv
name	–	conv1_1	relu1_1	conv1_2	relu1_2	pool1	conv2_1	relu2_1	conv2_2	relu2_2	pool2	conv3_1	relu3_1	conv3_2	relu3_2	conv3_3	relu3_3	pool3	conv4_1
support	–	3	1	3	1	2	3	1	3	1	2	3	1	3	1	3	1	2	3
filt dim	–	3	–	64	–	–	64	–	128	–	–	128	–	256	–	256	–	–	256
num filts	–	64	–	64	–	–	128	–	128	–	–	256	–	256	–	256	–	–	512
stride	–	1	1	1	1	2	1	1	1	1	2	1	1	1	1	1	1	2	1
pad	–	1	0	1	0	0	1	0	1	0	0	1	0	1	0	1	0	0	1

layer	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
type	relu	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	softmx
name	relu4_1	conv4_2	relu4_2	conv4_3	relu4_3	pool4	conv5_1	relu5_1	conv5_2	relu5_2	conv5_3	relu5_3	pool5	fc6	relu6	fc7	relu7	fc8	prob
support	1	3	1	3	1	2	3	1	3	1	3	1	2	7	1	1	1	1	1
filt dim	–	512	–	512	–	–	512	–	512	–	512	–	–	512	–	4096	–	4096	–
num filts	–	512	–	512	–	–	512	–	512	–	512	–	–	4096	–	4096	–	2622	–
stride	1	1	1	1	1	2	1	1	1	1	1	1	2	1	1	1	1	1	1
pad	0	1	0	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0

Ilustración 2.4 Detalles de la configuración de la CNN.

Comparación de características

La comparación entre vectores de características de dos imágenes se lleva a cabo mediante la similitud coseno.

$$similitud = \frac{A \cdot B}{||A||\,||B||} = \frac{\sum_{i=1}^{4096} A_i B_i}{\sqrt{\sum_{i=1}^{4096} A_i^2}\ \sqrt{\sum_{i=1}^{4096} B_i^2}}$$

donde A_i, B_i, son los componentes de los vectores de características A y B extraídos de dos caras a través de VGG-Face respectivamente.

Para la similitud coseno, a mayor valor obtenido en la comparación, mayor será la similitud entre ambas imágenes comparadas.

2.3.2 Caso de uso A: Buscador de similitudes

Se toma la imagen de entrada y se detectan todas las caras presentes en dicha imagen. De entre todas las caras detectadas en la imagen, se queda solamente con aquella que considera más importante. Para escoger la cara más importante tiene en cuenta dos factores: el tamaño de la cara detectada dentro de la imagen y la cercanía de la cara al centro de la imagen. Una vez obtenida la cara, la compara con las caras de las imágenes de los sujetos presentes en la base de datos y, en tiempo real, devuelve una lista ordenada con los valores de las comparaciones de las imágenes que han conseguido una mayor similitud con la persona presente en la imagen de entrada. Además de devolver esta lista, devuelve las identidades asociadas a dichos valores. Con esto se tienen las identidades de la base de datos colocadas de mayor a menor similitud con la imagen original. La aplicación permite demostrar la capacidad de la red para encontrar similitudes en caras que no son de la misma persona. Esto permite dar a conocer los atributos característicos (e.j. género, edad, barba, gafas, etc…) que son modelados a través de estas arquitecturas.

2.3.2.1 Base de datos utilizada

Se ha generado una base de datos a partir de las imágenes de los actores y actrices más famosos presentes en la página web https://www.imdb.com/. La información y las imágenes se han extraído de la página web mediante web scraping.

La base de datos cuenta con los nombres e imágenes de un total de 128.863 personalidades. Por cada individuo se tiene una sola imagen. En la ilustración 2.5 se pueden observar algunos ejemplos de las imágenes de la base de datos usada en este caso de uso.

Ilustración 2.5. Ejemplos de imágenes de caras de la base de datos utilizada.

En la ilustración 2.6 se pueden observar algunas imágenes de la base de datos original y sus versiones segmentadas en la nueva base de datos. La región facial extraída por el detector es normalizada a un tamaño fijo de 224×224 píxeles.

Ilustración 2.6. Segmentación de las imágenes de la base de datos original para la creación de la nueva base de datos con la que se ha trabajado.

2.3.2.2 Proceso del buscador de similitudes

Se ha separado el proceso en dos partes para disminuir la repetición de lecturas de los datos de la base de datos y así aumentar la eficiencia del algoritmo de reconocimiento.

En la primera, se hace un procesado de los datos de la base de datos previamente a la ejecución del algoritmo de reconocimiento y se toman los datos ya procesados al querer trabajar con la información de la base de datos.

Las caras son detectadas a partir del detector facial visto en apartados anteriores. Se extrae un vector de características (capa fc6 de VGG-Face) por cada cara y se almacena en una base de datos en forma de matriz que contiene los 128.863 vectores.

Finalmente, se crea de un diccionario de características que relacione los nombres de las imágenes de la nueva base de datos con los vectores de características que les correspondan, como se muestra en la ilustración 2.7.

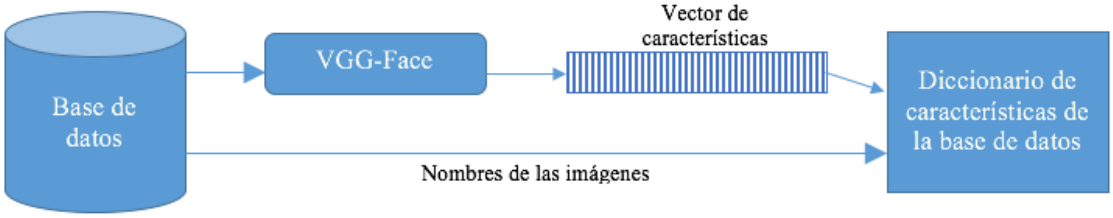

Ilustración 2.7. Primera parte del proceso: Creación del diccionario con la información de los vectores de características de las caras detectadas en la base de datos.

En la segunda parte del proceso, se llevan a cabo los pasos propios del algoritmo de reconocimiento facial. Este proceso se representa en la ilustración 2.8.

Se coge la imagen que se tiene de la interfaz desarrollada (webcam o fichero) y sobre ella, detecta todas las caras presentes. De entre todas las caras detectadas solamente se queda con aquella que considere que tiene más importancia. Determina la cara con mayor importancia teniendo en cuenta información del tamaño de la imagen y de la cercanía de la cara al centro de la imagen.

Finalmente, devuelve las cuatro coordenadas que marcan el cuadrado que delimita la cara detectada.

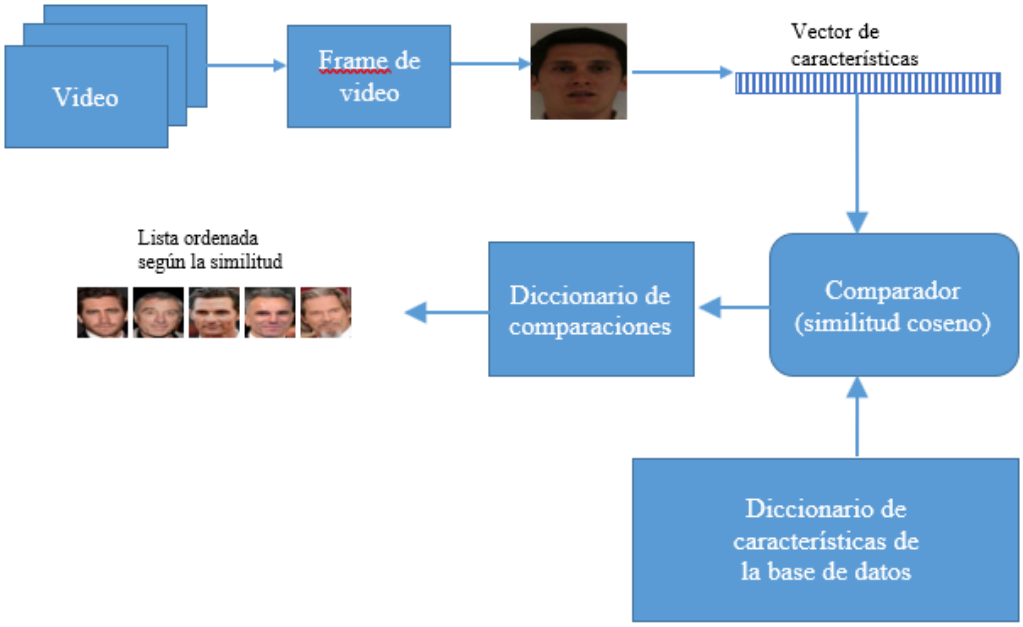

Ilustración 2.8. Segunda parte del proceso del caso de uso buscador de similitudes.

Con el objetivo de mejorar el reconocimiento, se le añaden unos márgenes el área cubierto por estas coordenadas, de este modo se consigue la información lo más completa posible, mediante la introducción de otros elementos, como el pelo o la forma completa de la cara. El tamaño asignado a los márgenes depende del tamaño de la cara dentro de la imagen.
A partir de las nuevas cuatro coordenadas, hace la extracción de la cara detectada. Para ello realiza un recorte de la imagen original en el área contenido por esas coordenadas.

Los conjuntos de cuatro coordenadas de las caras detectadas se utilizan también para modificar la imagen original añadiéndole la información visual de las localizaciones de las caras detectadas mediante la representación del cuadrado delimitante.

A continuación, obtiene el vector de características de la cara segmentada mediante el extractor de características. Se pasa este vector de características de la imagen junto con el diccionario de características de la base de datos al algoritmo de identificación.

En el algoritmo de identificación, dentro del diccionario, se recorren los vectores de características de las imágenes pertenecientes a un mismo usuario y se comparan con el vector de características de la imagen de entrada. De entre los valores de similitud obtenidos, se guarda para ese usuario el mayor valor. Una vez realizado este proceso para todos los usuarios presentes en la base de datos, se seleccionan los valores más altos y se devuelve una lista ordenada según el valor de la similitud con la imagen de entrada, junto con las identidades asociadas a dichos valores.

Finalmente, la identidad que encabeza la lista de similitud también se añade visualmente mediante la modificación de la imagen original.

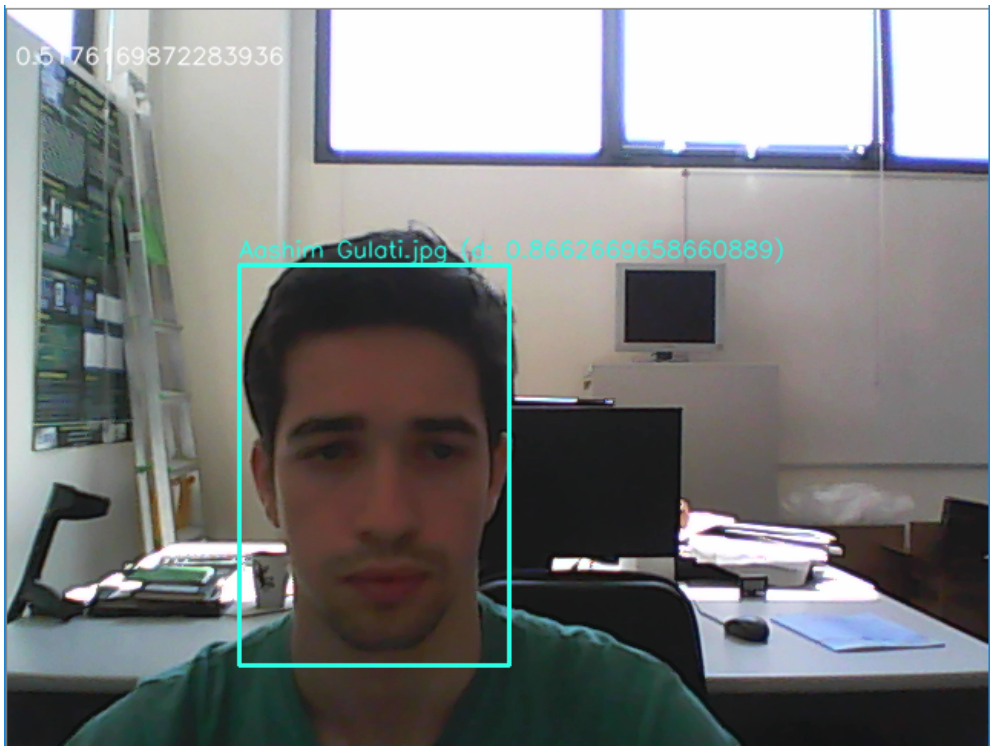

Ilustración 2.9 Resultado del algoritmo de detección, extracción e identificación de caras.

En la ilustración 2.9 se pueden observar los cambios visuales que añade este algoritmo sobre la imagen original. La representación de la imagen, no se produce en este algoritmo, sino al final del caso de uso, cuando devuelve el conjunto de los frames al front-end del demostrador.

2.3.3 Caso de uso B: Identificador facial

Se coge la imagen que se tiene de la interfaz desarrollada (webcam o fichero) y se detectan todas las caras presentes en ella. Para cada una de las caras detectadas en la imagen de entrada, compara la información de la cara, su vector de características, con la información de cada una de las caras presentes en las bases de datos. Una vez terminadas las comparaciones informa de si las caras detectadas en los frames corresponden a alguno de los usuarios presentes en la base de datos. En caso de corresponder, devuelve la identidad de los usuarios a los que corresponden, si por el contrario no corresponden a ninguno de los usuarios, informa de que se trata de una persona desconocida. Esta información de la localización e identidad de las caras detectadas se incrusta visualmente mediante la modificación de la imagen original.

2.3.3.1 Base de datos utilizada

Se tiene una base de datos de pequeñas dimensiones con imágenes capturadas en entornos controlados, entendiéndolo así por el hecho de que los sujetos cooperaban a la hora de la creación de las capturas, pero con una serie de problemas típicos de los entornos no

controlados, como diferencias en pose, iluminación, resolución, etc. El demostrador permite añadir identidades a través de la copia de imágenes etiquetadas en un directorio específico.

La base de datos usada en este caso de uso tiene la particularidad de aceptar más de una imagen por usuario y el número de imágenes por usuario no está definido, puede variar para cada usuario. Esto hace que se tenga que trabajar de diferente manera a la hora de guardar y extraer la información de la base de datos.
A diferencia del caso de uso buscador de similitudes, cada vez que se ejecuta el demostrador se parametriza la base de datos en busca de nuevas identidades o imágenes añadidas.

2.3.3.2 Proceso del identificador facial

En este caso de uso, el volumen de datos de la base de datos es muy pequeño, por lo que se ha optado por incluir el procesado de los datos de la base de datos dentro del propio algoritmo de reconocimiento, en lugar de hacer un paso previo externo como en el caso de uso buscador de similitudes.

El procedimiento seguido es el siguiente:
Se recorren las imágenes de la base de datos una a una. Para cada imagen detecta y extrae las caras presentes y guarda la más relevante. Por último, se obtienen los vectores de características de las caras mediante el extractor de características.

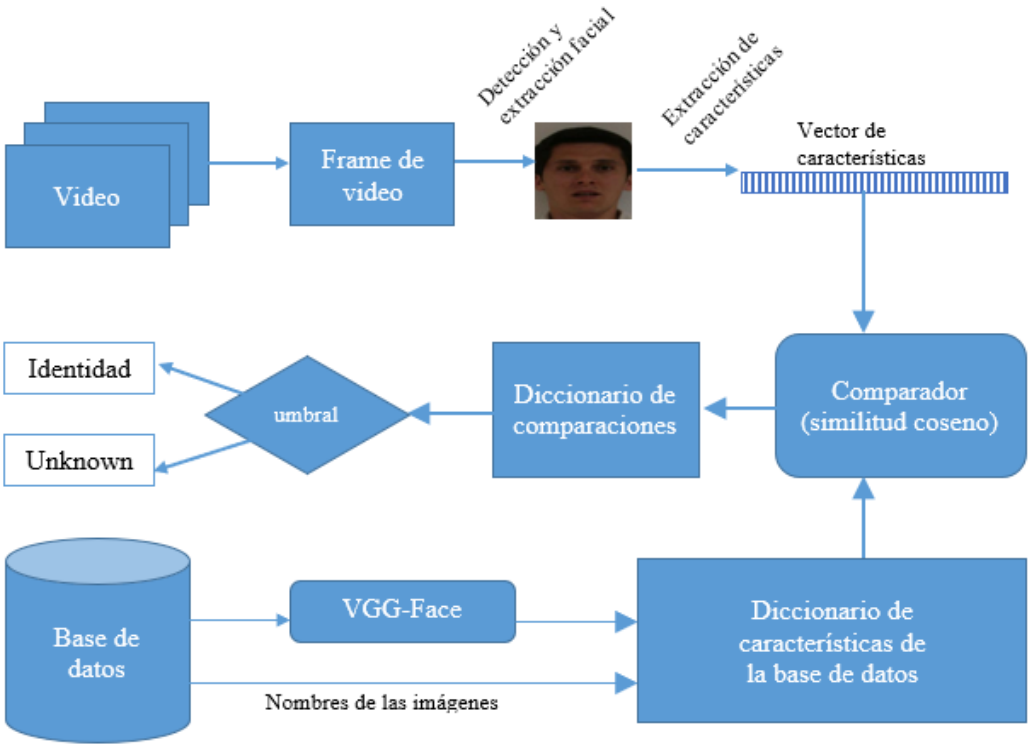

Ilustración 2.10. Proceso seguido para la implementación del caso de uso "Identificación facial".

El siguiente paso es la creación de un diccionario con el que poder acceder fácilmente a los vectores de características de la base de datos y a los nombres de las imágenes asociadas a dichos vectores de características.

A continuación, se toma la imagen de entrada y se aplica sobre ella los algoritmos de extracción, detección e identificación de caras. El proceso de la extracción de los vectores de características de las caras de la imagen de entrada es el mismo que en el buscador de similitudes, pero en este caso se queda con todas las caras detectadas dentro de la imagen. Una vez obtenidos los vectores de características de las caras de la imagen de entrada, se hace la comparación entre los vectores de características de las caras detectadas con los vectores de características de las imágenes de la base de datos. Las comparaciones se hacen como se explica en el apartado 2.3.1. Del resultado de estas comparaciones, se queda con la imagen que obtuvo el mayor valor de similitud para cada cara. También se coge la identidad del usuario de la base de datos asociada a esa imagen. De este modo se obtienen los usuarios con mayor parecido a las caras presentes en la imagen de entrada.

Se fija un umbral de comparación con el objetivo de saber si la imagen de la base de datos que consigue la máxima similitud corresponde al mismo individuo presente en la imagen de entrada o si simplemente se trata de la imagen con mayor similitud y el individuo de la imagen de entrada no se encuentra en la base de datos siendo una persona desconocida para el sistema. Si la similitud obtenida de la comparación supera el valor del umbral, la identidad de ambas imágenes coincide, se dará por verificada. Si el valor de la comparación es menor a dicho umbral, se devolverá la etiqueta "desconocido". Esto se puede observar en la ilustración 2.10.

A diferencia del caso de uso A, al final del proceso, no devuelve una lista ordenada de los valores de similitud, sino que devuelve la identidad del usuario de la base de datos que tiene una mayor similitud con la cara de la imagen de entrada. Este es el usuario identificado.

Por último, se añade la información obtenida de forma visual mediante la modificación de la imagen de entrada como se puede ver en la ilustración 2.11.

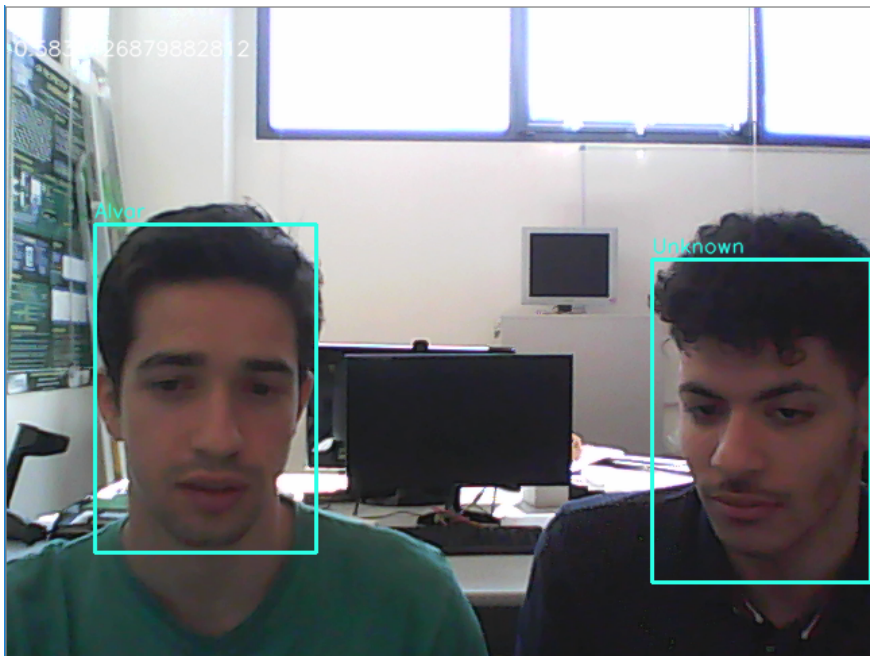

Ilustración 2.11 Resultados visuales añadidos sobre la imagen capturada original.

3 Mejoras en reconocimiento facial en entornos no controlados

3.1 Base de datos utilizada

El caso de reconocimiento facial en entornos no controlados tratado en este TFG está enfocado más específicamente al caso de reconocimiento de sujetos en imágenes de videovigilancia, aunque los resultados sean relevantes en cualquier entorno no controlado. Por tanto, las bases de dato con las que se trabaja contienen imágenes capturadas en entornos no controlados de videovigilancia sin la colaboración de los sujetos.

Se hará uso de una base de datos pública compuesta por tres conjuntos de imágenes de 90 usuarios diferentes. Más concretamente, se hará uso de la base de datos QUIS-CAMPI propuesta en [21]. En la figura 3.1 se pueden observar ejemplos de la base para dos usuarios diferentes.

Ilustración 3.1 Imágenes pertenecientes a los conjuntos de galería y evaluación para los usuarios 1 y 2 de la base de datos QUIS-CAMPI.

Los diferentes conjuntos en los que se divide la base de datos son:

- Imágenes de galería – Imágenes tomadas en circunstancias óptimas, con las que se entrena el algoritmo de reconocimiento. Contiene 3 imágenes por usuario, que corresponden a una imagen frontal, una imagen del perfil derecho y otra imagen del perfil izquierdo del sujeto. Todas las imágenes son de cuerpo entero.
- Imágenes de prueba – Ayudan a modelar a los usuarios en condiciones más similares a las de evaluación. Contiene 5 imágenes por usuario, imágenes captadas por cámaras de videovigilancia.
- Imágenes de evaluación – Imágenes sobre las que probaremos la eficiencia de los diferentes algoritmos desarrollados. Contiene 5 imágenes por usuario, imágenes captadas por cámaras de videovigilancia.

No se trabaja sobre las imágenes de la base de datos original directamente. Se crea una nueva base de datos a partir de la original. Esta nueva base de datos consiste en segmentaciones de las imágenes originales en el área en el que en se localizan las caras. Es el resultado de la aplicación del algoritmo de detección facial [20] sobre las imágenes de las bases de datos originales.
Los resultados de estas segmentaciones se pueden observar en la ilustración 3.2.

Ilustración 3.2 Ejemplos de la segmentación de la base de datos original para la creación de la nueva base de datos.

En la ilustración 3.3 se pueden observar todas las imágenes de un mismo usuario para cada uno de los conjuntos de la nueva base de datos.

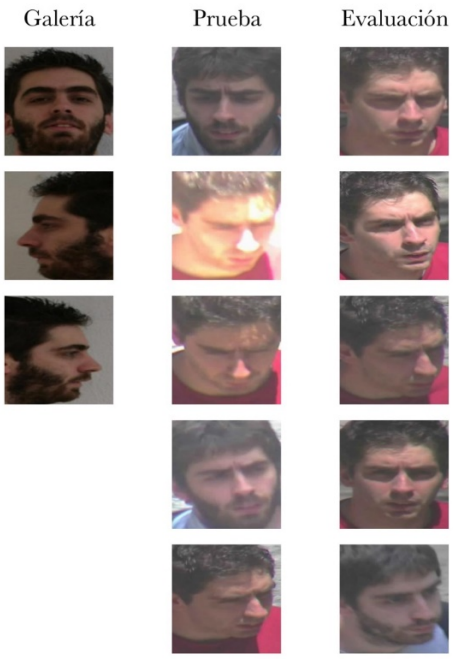

Ilustración 3.3 Imágenes segmentadas de la base de datos QUIS-CAMPI para el usuario 17 en los tres conjuntos de la base de datos.

3.2 Selección de características dependientes de usuario

En este apartado se presentan las diferentes hipótesis con las que se tratará de mejorar la eficiencia de los algoritmos de reconocimiento facial del estado del arte.

Se sabe que, dentro de la base de datos con la que contamos, el conjunto de entrenamiento (galería) incluye un conjunto de imágenes para cada uno de los usuarios presentes en la base de datos. Se tratará de seleccionar la información relevante a nivel de usuario a partir del conjunto de imágenes para cada usuario. Para ello, se llevará a cabo una selección de las características más estables para los vectores de características extraídos de las imágenes de un mismo usuario.

Una vez se ha detectado la cara, se extrae la región de interés y mediante el extractor de características explicado en la sección 2.3 Casos de uso, se obtiene un vector de características que representa la información contenido en el área en la que se detectó la cara. El vector de características obtenido contiene 4096 características.

Del conjunto de 4096 características de cada imagen, se tratará de averiguar cuáles son las características más importantes a la hora de reconocer a cada usuario. Asumiremos que hay un subconjunto de los vectores de características de las imágenes de un mismo usuario que tiene información relevante a la hora de reconocer a ese usuario del resto de usuarios. De forma intuitiva, se aplicará una selección de características dependiente de usuario. Esta selección buscará características que sean robustas en entornos no controlados donde las distorsiones son severas y la perdida de rendimiento de los sistemas de reconocimiento facial clara.

La información que se tiene en cuenta a la hora de encontrar el subconjunto de características y hacer la selección de características será la del conjunto de galería, conjunto cuyas

imágenes han sido tomadas en entornos controlados. Se parte por tanto de tres imágenes adquiridas en condiciones muy diferentes a las que se tendrán que enfrentar posteriormente como prueba o test.

La selección de características propuesta se basa en explotar la estabilidad de las características extraídas por la red convolucional. La hipótesis de partida es que algunas de las 4096 características serán más robustas a distorsiones de pose, oclusión o distancia entre otras. Hay que tener en cuenta que en el conjunto de galería disponemos de tres poses diferentes, lo que nos permitirá seleccionar características invariantes a la pose (una de las principales distorsiones en entornos no controlados como se puede observar en la ilustración Ilustración 1.1).

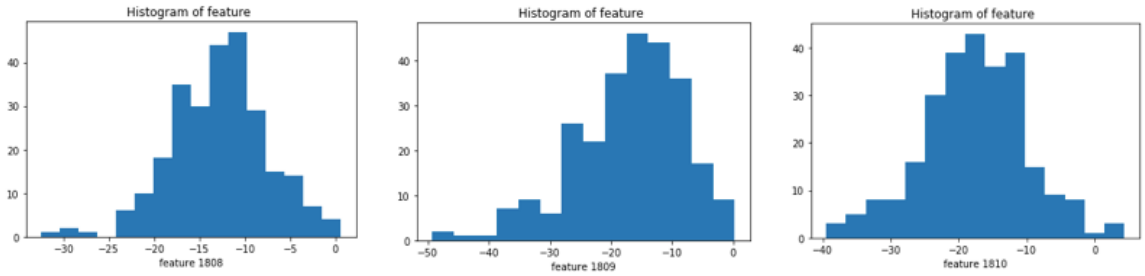

Ilustración 3.4 Representación de las distribuciones de los valores para tres diferentes características de entre las 4096 presentes en los vectores de características.

Se ha podido comprobar que los valores de cada característica del vector de características para el conjunto de las imágenes de galería suelen seguir una distribución normal, como se puede observar en la ilustración 3.4. Sabiendo esto, la selección de características para un usuario se hará mediante la comparación de las distribuciones de las características para todo el conjunto de imágenes de galería con los valores de las características de las imágenes para ese mismo usuario. Se buscarán aquellas características estables dentro de las imágenes disponibles de un mismo usuario.

Una vez obtenido el subconjunto de las características relevantes para el reconocimiento de un usuario, se creará una máscara binaria para ese usuario. Cada valor de la máscara corresponderá a una de las características de los vectores de características. Si una característica se ha definido como relevante para el reconocimiento de un usuario, se le asignará un 1, y en caso de no ser relevante, se le asignará un 0. Estas máscaras obtenidas se aplicarán sobre los vectores de características de las imágenes a la hora de hacer las comparaciones de similitud entre imágenes.

En los siguientes apartados se explicará cada una de las diferentes hipótesis probadas a la hora de crear estas máscaras de selección de características y se darán los resultados obtenidos para cada una hipótesis.

Sabiendo como se encuentran distribuidas las imágenes en las bases de datos, se han tratados dos casos diferentes:

- Galería contra evaluación, en el que solo se dispone de las imágenes obtenidas en un entorno controlado a la hora de obtener la selección de características y crear las plantillas de usuarios.
- Galería y prueba contra evaluación, en el que además se añade la información de las imágenes de prueba para modelar mejor a los usuarios. Se dispone por tanto de

imágenes adquiridas en condiciones similares a las de evaluación para crear las plantillas de usuario.

3.2.1 Galería contra evaluación: no se dispone de imágenes de prueba

A la hora de hacer la selección de características y la creación de las máscaras, solamente se cuenta con la información del conjunto de galería, cuyas imágenes han sido tomadas en condiciones controladas.

Dado un conjunto de vectores de características $f_i^{frontal}$, $f_i^{perfilD}$, $f_i^{perfilI} \in \mathbb{R}^{4096\times1}$ obtenidos de la imagen frontal, de la imagen del perfil derecho y de la imagen del perfil izquierdo del usuario i, se genera un conjunto de *binary embeddings* $b_i^{frontal}, b_i^{perfilD}, b_i^{perfilI} \in \mathbb{R}^{4096\times1}$ para las imágenes del usuario. El proceso de creación de los binary embeddings varía según la hipótesis, por lo que se explica en cada una de ellas el proceso a seguir.

Habiendo obtenido las *binary embeddings* para las imágenes de un usuario i, se plantean diferentes métodos para la obtención de las máscaras de usuarios $m_i^{usuario} \in \mathbb{R}^{4096\times1}$ para el usuario i basándonos en la posición de las tres características j disponibles para cada usuario, así como la distribución de probabilidad $g(j)$ de esa característica para todas las muestras de entrenamiento del resto de usuarios. Se considerará que una característica es estable, cuando las muestras se encuentren cercanas entre si dentro de la distribución de probabilidad $g(j)$. El método para la comprobación de esta cercanía o el grado de cercanía dependerá en cada hipótesis propuesta.

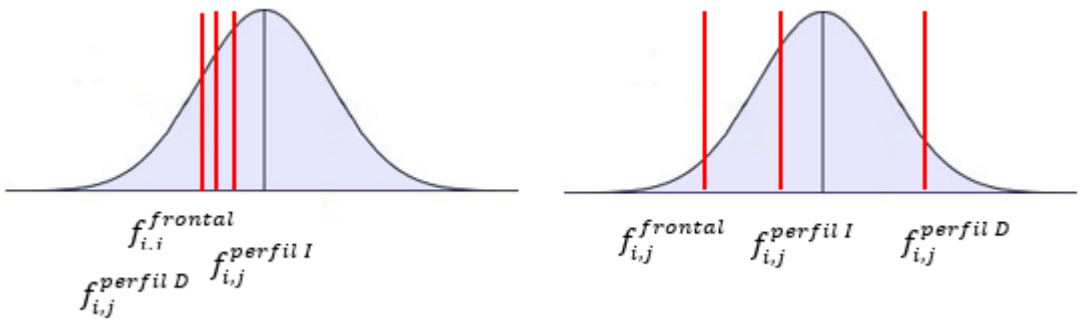

Ilustración 3.5 Distribuciones de probabilidad g de una característica j para las tres imágenes de galería de un usuario i. A la izquierda caso en el que las tres características se consideran estables. A la derecha caso en el que no se consideran estables.

Se crea una matriz $M_{i,j}$ *donde* $i \in [1{,}270]$ corresponde a las imágenes en el conjunto de galería (3 imágenes para cada uno de los 90 usuarios) y $j \in [1, 4096]$ a las características en los vectores de características. Esta matriz contiene la información de todos los vectores de características correspondientes a las imágenes del conjunto de galería.

Los desarrollos de las hipótesis planteadas siguen un procedimiento común que se puede observar en la ilustración 3.5, y que a niveles prácticos está separado en dos partes:

- El cálculo de las *binary embeddings* a partir de la información de las imágenes, información contenida en la matriz $M_{i,j}$. Más concretamente se calcula una *binary embedding* por imagen.
- El cálculo de las máscaras de usuario. Una vez obtenidas las *binary embeddings*, se agrupan según el usuario al que pertenecen y se buscan aquellas características estables dentro de las *binary embeddings* de un mismo usuario según un criterio que dependerá de la hipótesis en la que se encuentre. Con esta información, se obtiene una máscara para ese usuario.

Es importante entender que las *binary embeddings* no son máscaras, no tienen como objetivo marcar la relevancia de las características a la hora de reconocer imágenes. Son un conjunto de vectores de valores binarios. En su creación se les ha asignado el valor 1 o 0 a cada uno de sus valores en función de cumplir ciertas propiedades dentro de sus distribuciones de características. La propiedad que debe cumplir una característica del vector de características para que se le asigne 1 dependerá de la hipótesis en la que se encuentre.

Estos vectores se han creado con la intención de simplificar la información contenida en los vectores de características mediante una clasificación de sus características en dos conjuntos, 1 o 0, en función de una particularidad escogida. Esta simplificación de las características hace que la creación de las máscaras de usuarios sea mucho más sencilla.

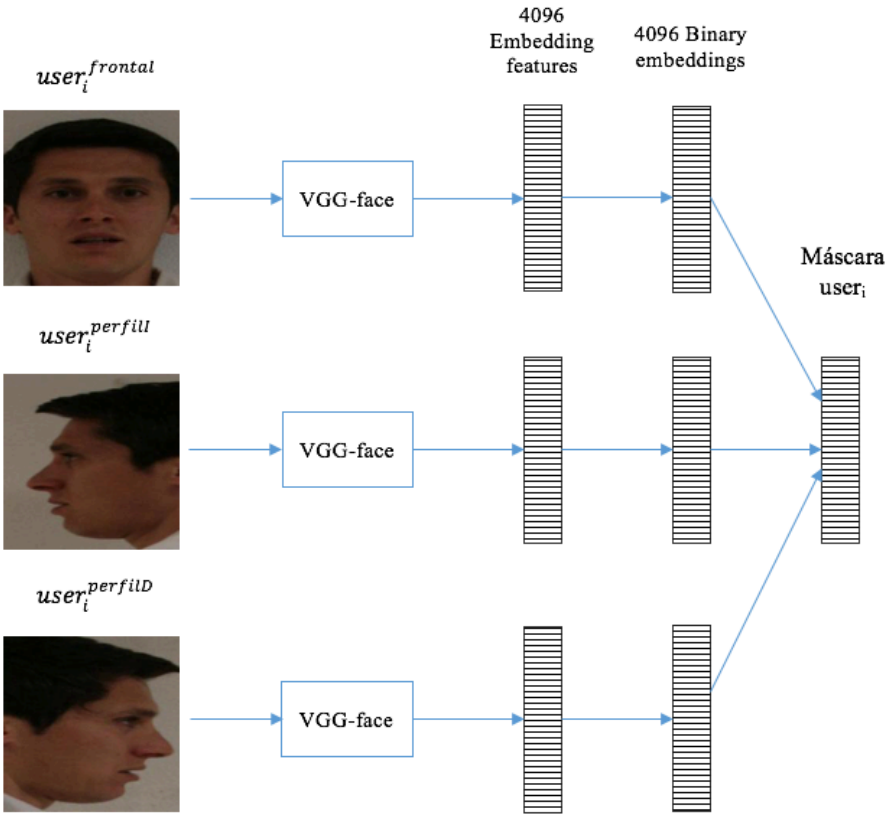

Ilustración 3.6 Proceso seguido para la creación de las máscaras de usuario en las hipótesis planteadas.

Los desarrollos de las diferentes hipótesis se centran por tanto en dos aspectos:

- la particularidad tenida en cuenta a la hora de la creación de las *binary embeddings*,

- la relación que se busca que cumplan los diferentes valores de las *binary embeddings* para las imágenes de un mismo usuario, a la hora de la creación de las máscaras de usuarios.

3.2.1.1 XNOR por usuario

Obtención de las *binary embeddings*:
Teniendo la matriz de características $M_{i,j}$, se toman una a una las columnas que la forman. Estas columnas corresponden a los valores de una característica para todas las imágenes de la base de datos. Para cada característica (columna de $M_{i,j}$), se obtiene su distribución de probabilidad $g(j)$, distribución estadística del valor de la característica para todas las imágenes del conjunto, y se calcula el valor de la media aritmética de los valores de $g(j)$.

Para una imagen m, se compara uno a uno el valor de cada una de las características $j \in [1, 4096]$, con la media de los valores de $g(j)$. En aquellos casos en los que el valor de la característica es menor para la imagen que para la media de la característica, se le asigna un 1 en la máscara; en aquellos casos que el valor para la imagen sea superior a la media, se le asigna un 0.

Obtención de las máscaras de usuarios:
Una vez obtenidos las *binary embeddings* de las imágenes de un usuario i: $b_i^{frontal}$, $m_i^{perfilD}$, $m_i^{perfilI}$, se calcula las máscara del usuario i mediante la aplicación de la XNOR sobre sus *binary embeddings*, $m_i^{xnor} = \text{xnor}(m_{i,j}^{frontal}, m_{i,j}^{perfilD}, m_{i,j}^{perfilI})$ $para\ j \in [1, 4096]$.

Esta hipótesis se basa en la búsqueda de las características estables para un mismo usuario según lo planteado en la ilustración 3.6.

Posibles problemas de este método:
Al estar basado en la separación de valores según la media, puede haber valores próximos dentro de la distribución que sean asignados como no relevantes para el reconocimiento del usuario debido a estar muy cerca de la media, pudiendo tener alguno de ellos un valor superior o inferior a la media teniendo el resto de los valores de las máscaras el valor contrario.
Además, esto se potencia debido a que es justo alrededor de la media en donde mayor concentración de valores tenemos al tratarse de una distribución normal.

3.2.1.2 AND por usuario

Obtención de las máscaras de imágenes:
Dentro de esta hipótesis va a haber cuatro casos diferenciados a la hora de la obtención de las máscaras de imágenes dependiendo de los diferentes valores con los que se quedará dentro de la función de distribución.

El procedimiento es muy similar al seguido en la hipótesis XNOR por usuario para todos los casos, que se pueden observar en la ilustración 3.7:

- **Caso A**
  Se asignan a 1 aquellas características con valores inferiores a un umbral, umbral cuyo valor corresponde a diferentes variaciones del valor de la media. Se anulan aquellas características que tengan valores en la cola superior de la distribución. Con esto, se eliminan los valores más altos y con menos probabilidad de ocurrencia.
- **Caso B**
  Se asignan a 1 aquellas características con valores superiores a un umbral. Se anulan aquellas características que tengan valores en la cola inferior de la distribución.
- **Caso C**
  Se toman como relevantes aquellas características con valores pertenecientes a ambas colas dentro de la distribución de la media. Conociendo los valores de la distribución, se definen 2 umbrales. Se crea una primera máscara en la que se asigna 1 a aquellas características inferiores al umbral inferior. Se crea una segunda máscara en la que se asigna 1 a aquellas características superiores al umbral superior. Las máscaras de imágenes se consiguen mediante la unión de estas dos máscaras. En la implementación simplemente se seleccionan los valores opuestos a los obtenidos en el caso C.

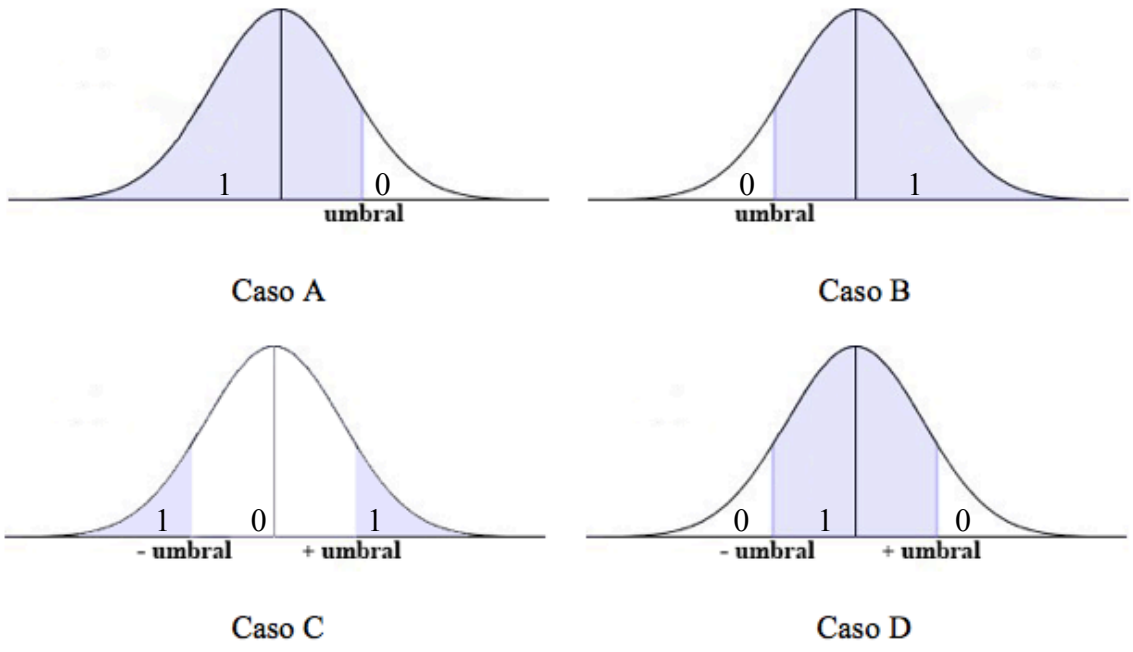

Ilustración 3.7 Diferentes valores tomados de la distribución de probabilidad de los valores de una característica para la creación de las máscaras de imágenes.

- **Caso D**
  Se toman como relevantes aquellas características con valores centrales dentro de la distribución de la media. Conociendo los valores de la distribución, se definen 2 umbrales. Se crea una primera máscara en la que se asigna 1 a aquellas características inferiores al umbral superior. Se crea una segunda máscara en la que se asigna 1 a aquellas características superiores al umbral inferior. Las máscaras de imágenes se consiguen mediante la intersección de estas dos máscaras.

Obtención de las máscaras de usuarios:

Una vez obtenidos las *binary embeddings* de las imágenes de un usuario i: $m_i^{frontal}$, $m_i^{perfilD}$, $m_i^{perfilI}$, se calcula las máscara del usuario i mediante la aplicación de la AND sobre sus *binary embeddings*, $m_i^{and} = \text{and}(b_{i,j}^{frontal}, b_{i,j}^{perfilD}, b_{i,j}^{perfilI})\ para\ j \in [1, 4096]$.

3.2.1.3 Varianza de usuario

Sabiendo que las distribuciones para las características se asemejan a una distribución normal, se hace la selección de características en función de la varianza de la característica. Para ello, para cada característica j, se tendrá en cuenta la varianza de la característica para las imágenes de un mismo usuario i, $(\sigma^2)_i^j$; y la varianza de la característica para el total de las imágenes de la base de datos $(\sigma^2)_{total}^j$.

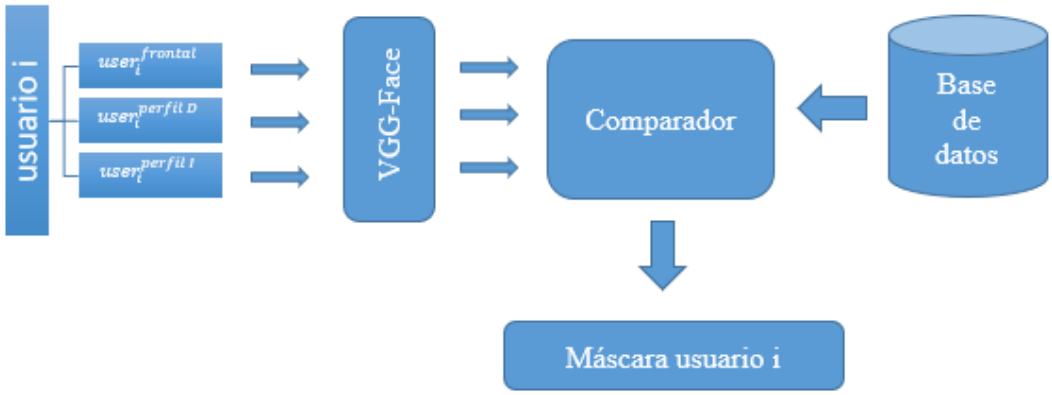

Ilustración 3.8 Proceso seguido para la creación de las máscaras de usuario en las hipótesis Varianza de usuario.

Obtención de las máscaras de usuarios:

En este caso no hay cálculo de las máscaras de imágenes, se calculan las máscaras de usuarios directamente.

El primer paso es calcular la varianza total $(\sigma^2)_{total}^j$ para cada característica j, cada columna de la matriz $M_{i,j}$. El segundo paso es calcular la varianza de las características para cada usuario i donde $i \in [1, 90]$. Se agrupan los vectores de características de las imágenes correspondientes a ese mismo usuario i. Para cada característica $j \in [1, 4096]$, calculamos $(\sigma^2)_i^j$, las varianzas de cada una de las características para los vectores de características de las imágenes del usuario i.

Se toman como relevantes para el usuario i, asignadas a 1 en la máscara del usuario i, aquellas características para las que el valor de la varianza del usuario i, σ_i^2, sea menor que un porcentaje de la varianza total de la característica, σ_{total}^2. Ese porcentaje es un valor prefijado.

3.2.2 Galería y prueba contra evaluación

En este caso se cuenta con las imágenes del conjunto de prueba y del conjunto de galería a la hora de hacer la selección de características y la creación de las máscaras de usuarios. Por tanto, sabiendo que se sigue el mismo procedimiento para la implementación de las hipótesis, el primer punto es la unión de los datos de los conjuntos de galería y prueba.

Es importante que la unión de los datos de prueba y de galería se realice antes del cálculo de las máscaras, de este modo, al hacerse las comparaciones de las características para las imágenes de un mismo usuario con las características de todas las imágenes de ambos conjuntos de la base de datos, se tiene en cuenta la información compuesta por ambos conjuntos. En caso de hacerse los cálculos de las máscaras de imágenes para los datos de cada conjunto separadamente y unirse una vez obtenidos ambos conjuntos de máscaras, se modificarían los resultados, al no haberse realizado los cálculos sobre el total de los datos.

El procedimiento para la obtención de las máscaras de imágenes teniendo en cuenta los datos de los conjuntos de galería y prueba es el siguiente:

- Se crea una lista común con los nombres uniendo las listas de nombres de las imágenes de galería y las de prueba.
- Se calculan los vectores de características de las imágenes de cada base de datos y se crea una matriz que contenga la información de los vectores para cada conjunto de la base de datos. Se crea una matriz concatenando las 2 matrices de características creadas con los datos de los conjuntos de galería y prueba.
- Se calculan las *binary embeddings* a partir de esta nueva lista y esta nueva matriz. El procedimiento para los cálculos de las máscaras de imágenes dependerá de la hipótesis en la que se encuentre.

Las hipótesis “XNOR por usuario”, “AND por usuario” y “Varianza de cada usuario” siguen un procedimiento idéntico al ya explicado para estas mismas hipótesis en 3.2.1 Galería contra evaluación. La única diferencia es que en este caso se ha unido la información de las características de galería y prueba antes del cálculo de las *binary embeddings*.

3.2.2.1 XNOR por usuario teniendo en cuenta la mayoría

Cálculo de las máscaras de imágenes

El procedimiento es idéntico al seguido en 3.2.1.1 XNOR por usuario, con la única diferencia de que en este caso se ha unido la información de las características de galería y prueba antes del cálculo de las máscaras de imágenes.

Cálculo de las máscaras de usuarios

En este caso en lugar de aplicar una XNOR por cada usuario con las máscaras correspondientes a sus imágenes, se aplica una XNOR en aquellos casos en los que se cumpla para la mayoría, así se evita que posibles datos aislados nos anulen una característica que podría ser relevante a la hora de reconocer al usuario.

Dado que para la unión de los conjuntos de galería y prueba se tiene un total de 8 imágenes por usuario, tendremos en cuenta aquellos valores que cumplan una condición preestablecida para 6 o más imágenes.

No se implementa directamente una XNOR, sino que se le asigna 1 en la máscara a aquellas características cuya suma sea inferior a 2 o superior a 6. Se le da el valor de dos de las máscaras como margen de error.

Esto se puede implementar de este modo gracias a que en el cálculo de las máscaras de imágenes ya se ha hecho una simplificación binaria de los valores de las características en función de si cumplían o no cierta condición.

3.3 Protocolo experimental

En los apartados anteriores se han calculado las máscaras de usuarios con las que se hace la selección de características sobre los vectores de características de las imágenes que se van a comparar, con el objetivo de mejorar el rendimiento en reconocimiento de usuarios.

En este apartado se va a explicar cómo calcular los resultados para el caso de referencia o baseline en el que no se aplica selección de características; y para los casos en los que se aplican las diferentes máscaras de usuarios. De este modo, se podrá comprobar si hay alguna mejora en el rendimiento al adaptar los modelos a los usuarios mediante la aplicación de estas máscaras.

3.3.1 Cálculos para la obtención de los scores en el caso baseline

Se va a explicar el proceso para el caso en el que se tiene las imágenes del conjunto de galería contra las del conjunto de evaluación. En el caso de tener los conjuntos de galería y prueba contra el conjunto de evaluación, se deben unir los datos de galería y prueba como se explicó en 3.2.2 Galería y prueba contra evaluación. A continuación, se usaría ese conjunto de datos tal y como aquí se van a usar los datos del conjunto de galería.

El proceso es el siguiente:

Se recorren las imágenes de evaluación, y para cada una de ellas buscamos a qué usuario pertenecen. En esta base de datos es una tarea sencilla dado que ha sido construida de tal modo que el nombre de la imagen contenga la información del usuario al que pertenece.

Se compara cada imagen de evaluación con todas las imágenes de galería. Para ello, para cada imagen de evaluación se recorren todas las imágenes de galería y del mismo modo que se ha hecho para la imagen de evaluación, buscamos a qué usuario pertenece la imagen de galería correspondiente.

Se crean dos diccionarios en los que se guardará la información obtenida en las comparaciones entre imágenes. En un diccionario se guardará aquella información obtenida para los casos en los que la imagen de galería sea genuina, los usuarios de la imagen de evaluación y la imagen de galería son el mismo. En el otro diccionario se guarda la información obtenida en los casos en los que la imagen de galería sea impostora, los usuarios no coinciden. De este modo podremos obtener la información correspondiente a los genuinos y a los impostores.

A continuación, se compara el vector de características de la imagen de evaluación con todas las imágenes de galería de un mismo usuario y para ese usuario, nos quedaremos con el mayor resultado obtenido. Por último, se guarda el resultado para esa imagen de evaluación y ese usuario de galería en el diccionario que corresponda.
Una vez terminado el proceso para todas las imágenes de evaluación, tendremos 2 diccionarios. El diccionario de los genuinos tendrá un total de 90x5x1 valores. El diccionario de los impostores tendrá un total de 90x5x89 valores. Mediante el uso de estos diccionarios se obtendrán los resultados de los falsos positivos y falsos negativos.

Se obtienen falsos positivos en aquellos casos en los que el reconocedor indica que se trata de un genuino y en realidad se trate de un impostor.
Se obtienen falsos negativos en aquellos casos que el reconocedor indica que se trata de un impostor cuando en realidad se trata de un genuino.

3.3.2 Cálculo de los scores en caso de haberse usado máscaras

Sigue un proceso idéntico al explicado en el apartado 3.3.1, pero antes de realizar la comparación entre los vectores de características de la imagen de evaluación y de una de las imágenes de galería, se aplica la máscara correspondiente sobre ambos vectores de características.

Teniendo una imagen de evaluación, correspondiente a un usuario i, y una imagen de galería, correspondiente a un usuario j, se aplica la máscara correspondiente al usuario i sobre los vectores de características obtenidos para ambas imágenes.

Las máscaras se han calculado sobre las imágenes del conjunto de galería (o de la unión de galería y prueba). Pero a la hora de aplicarlas, dado que queremos calcular los falsos positivos y falsos negativos para la imagen de evaluación, se aplicará la máscara correspondiente al usuario al que pertenece la imagen de evaluación.

3.4 Resultados

En esta sección se van a calcular los resultados tanto del caso original, al que hemos llamado baseline, como de los casos en los que se han aplicado las hipótesis planteadas previamente en los apartados 3.2.1 Galería contra evaluación y 3.2.2 Galería y prueba contra evaluación.

Para el cálculo de los resultados de eficiencia se hace uso de los siguientes valores:

La **False Acceptance Rate (FAR)** es la probabilidad de que un usuario impostor sea aceptado por el sistema como un usuario genuino.

La **False Rejection Rate (FRR)** es la probabilidad de que un usuario genuino sea rechazado por el sistema.

La **True Acceptance Rate (TAR)** es la probabilidad de que el sistema verifique como correcta la identidad de un usuario genuino.

$$TAR = 1 - FRR$$

La **Receiver Operating Curve (ROC)** es comúnmente usadas como herramienta gráfica para juzgar el rendimiento de los sistemas de autenticación. En ellas se presentan los valores de la FRR contra los valores de la FAR. En algunos casos también se pueden encontrar los valores de la TAR contra los de la FAR. En nuestro caso en particular, se representará la ROC como la TAR contra la FAR.

El **Equal Error Rate (ERR)** es el valor obtenido cuando FAR y FRR coinciden. En nuestro caso se encuentra representado como un porcentaje.

3.4.1 Resultados: Galería contra evaluación

Baseline	
Normalizado	**No normalizado**
16,25	17,76

XNOR	
Normalizado	**No normalizado**
16,81	21,80

AND, caso A										
Vector	**Umbral**	**1,0**	**0,8**	**0,4**	**0**	**-0,4**	**-0,6**	**-0,8**	**-1,0**	**-1,4**
Normalizado	**EER (%)**	17,30	16,86	15,99	15,05		13,25	12,85	12,63	12,59
No normalizado	**EER (%)**	14,38	13,54	13,25	13,38	11,46	11,23	11,21	11,70	12,35

AND, caso B				
Vector características	**Umbral**	**1,0**	**0,8**	**0,6**
Normalizado	**EER (%)**	17,75	17,60	17,15
No normalizado	**EER (%)**	25,97	28,28	31,01

AND, caso C			
Normalizado		**No normalizado**	
Umbral	**ERR (%)**	**Umbral**	**ERR (%)**
0.8	12,13	**0.8**	21,01

AND, caso D			
Normalizado		**No normalizado**	
Umbral	**ERR (%)**	**Umbral**	**ERR (%)**
0.8	33,24	**0.8**	21,61

Varianza					
Vector características	**Porcentaje**	**5%**	**10%**	**20%**	**35%**
Normalizado	**EER (%)**	18,87	18,04	17,75	17,52
No normalizado	**EER (%)**	25,71	24,04	22,77	21,35

Se ha conseguido una reducción del error de hasta un 31% en el rendimiento en aquellas hipótesis basadas en la eliminación de las características con valores que se encuentren dentro de la cola de superior de las distribuciones de las características, en aquellas hipótesis en los que se eliminan todos los valores a excepción de los presentes en la cola inferior y en

aquellas en las que solamente se queda con los valores presentes en ambas colas (señalado en verde). Los mejores resultados se obtienen en aquellos casos en los que solamente se seleccionan aquellas características cuyos valores se encuentran en la cola inferior de la distribución.

En cambio, para aquellas hipótesis basadas en la cercanía de las muestras dentro de la distribución de probabilidad no se han conseguido los resultados esperados, no consiguiendo mejoras en ninguna de estas hipótesis propuestas.

En la siguiente tabla y figura se encuentran representados los mejores resultados obtenidos para cada una de las hipótesis desarrolladas. El valor de la hipótesis de la varianza por usuario no se incluye dado que no mejora el rendimiento en ningún caso, solamente se va acercando al valor obtenido en el caso de referencia según se acerca el porcentaje al 100%, valor para el cual sería el mismo caso.

	Baseline	**XNOR**	**AND**
EER	16,25%	16,81% (↑3,5%)	11,21% (↓31%)

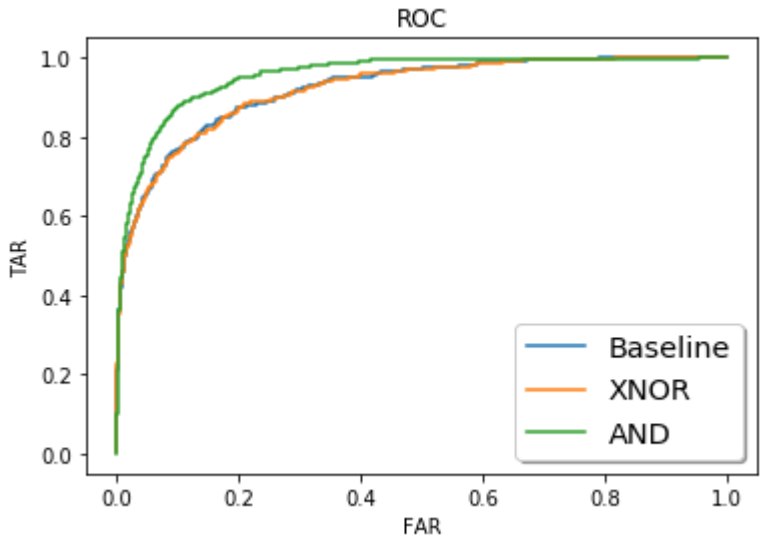

Ilustración 3.9 ROC para el caso baseline y para los mejores resultados obtenidos para cada uno de los valores

3.4.2 Resultados: Galería y prueba contra evaluación

Baseline	
Normalizado	**No normalizado**
4,71	4,39

XNOR	
Normalizado	**No normalizado**
4,92	4,60

AND, caso A								
Vector características	**Umbral**	**1,2**	**0,8**	**0,6**	**0,4**	**0**	**-0,8**	**-1,2**

Normalizado	**EER (%)**	4,71	4,71	5,16		5,16	4,71	
No normalizado	**EER (%)**	4,32	3,87	3,89	4,26	5,07	18,64	25,42

AND, caso B						
Vector características	**Umbral**	**2,0**	**1,6**	**1,2**	**0,8**	**0,6**
Normalizado	**EER (%)**		4,94	5,06	4,89	4,94
No normalizado	**EER (%)**	4,92	4,99	6,38	8,53	11,01

AND, caso C					
Vector características	**Umbral**	**1,0**	**0,8**	**0,6**	**0,4**
Normalizado	**EER (%)**	3,85	4,06	3,82	4,18
No normalizado	**EER (%)**	5,61	5,99	6,77	7,51

AND, caso D				
Vector características	**Umbral**	**1,2**	**0,8**	**0,6**
Normalizado	**EER (%)**	12,25	17,07	26,51
No normalizado	**EER (%)**	6,00	7,58	9,21

XNOR mayoría	
Normalizado	**No normalizado**
4,87	6,74

Varianza					
Vector características	**Porcentaje**	**5%**	**10%**	**20%**	**35%**
Normalizado	**EER (%)**	5,14	4,79	4,71	5,02
No normalizado	**EER (%)**	7,34	7,19	6,44	5,71

Los resultados obtenidos en este apartado arrojan las mismas conclusiones que en el apartado de galería contra evaluación.

En cualquier caso, las mejoras son menos significativas, alcanzando reducciones del error de un 13%. De cualquier modo, hay que tener en cuenta que en este caso el rendimiento para el caso de referencia alcanzaba un 95,61%, por lo que el margen de mejora era muy inferior al que nos encontrábamos en 3.4.1 Resultados: Galería contra evaluación.

En la siguiente tabla y figura se encuentran representados los mejores resultados obtenidos para cada una de las hipótesis desarrolladas. Por las razones explicadas en el apartado 3.4.1, no se incluyen los datos de la varianza.

	Baseline	**XNOR**	**AND**	**XNOR mayoría**
EER	4,39%	4,60% (↑4,7%)	3,82% (↓13%)	4,87% (↑11%)

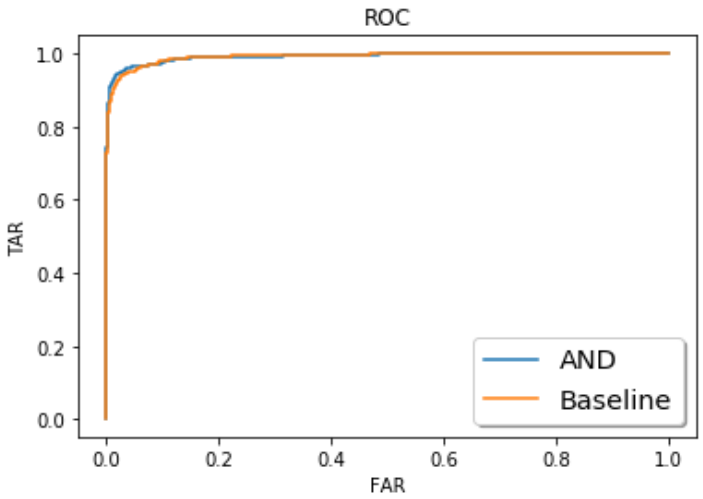

Ilustración 3.10 ROC para el caso baseline y para la hipótesis con la que se consigue la mayor reducción del error, la hipótesis AND.

En este caso en la ilustración 3.10 solamente se incluye el mejor resultado obtenido, el de la AND, ya que para el resto de los valores la diferencia es demasiado pequeña como para apreciarlos respecto del caso baseline.

4 Conclusiones y trabajo futuro

4.1 Conclusiones

Los algoritmos de reconocimiento facial están consiguiendo un extraordinario desempeño en los últimos años, pero en aquellos casos en los que trabajan con imágenes capturadas en entornos no controlados el rendimiento aún no alcanza los resultados deseados, habiendo aún un gran rango de mejora en este ámbito. Este caso es especialmente claro en el caso de la videovigilancia, en el que se potencian aún más los problemas que se plantean al trabajar con este tipo de imágenes al tener que lidiar con imágenes de baja calidad y grandes volúmenes de datos.

El gran desarrollo que ha habido en los últimos años en los sistemas de reconocimiento facial y el alto grado de integración que están teniendo en todos los ámbitos de la sociedad, así como el gran rango de mejora aún posible en aquellos casos en los que trabajan con imágenes captadas en entornos no controlados, han motivado los objetivos planteados en este Trabajo Fin de Grado.

Al comenzar el desarrollo de este TFG se partía con dos objetivos definidos: primero el desarrollo de una interfaz visual sobre la que fuera posible la evaluación de diferentes algoritmos de reconocimiento facial; y segundo, la mejora en el rendimiento de los sistemas de reconocimiento facial al trabajar con imágenes capturadas en entornos no controlados.

Se ha visto que es posible desarrollar un demostrador de tecnologías de reconocimiento facial en Python con rendimientos en el estado del arte. También se han dos algoritmos de reconocimiento que se ejecutan sobre la interfaz de este demostrador. El primero toma los frames del video capturado en el demostrador y localizando una lista de los usuarios de la base de datos que más se parecen a la persona presente en las imágenes capturadas. El segundo, busca en la base de datos la identidad de los individuos presentes en el video y determina si corresponden a usuarios de la base de datos o si son desconocidos para la base de datos.

En la segunda parte del TFG se ha podido demostrar que el rendimiento en entornos no controlados aún es mejorable. Se ha tratado de mejorar el rendimiento de los diferentes algoritmos de reconocimiento sobre imágenes capturadas en entornos no controlados, planteando una serie de hipótesis basadas en la selección de aquellas características más robustas dentro del conjunto de imágenes para un mismo usuario. Se ha demostrado que la mejora en este tipo de condiciones es posible, habiendo conseguido una mejora en el rendimiento del 31% respecto del caso de referencia del que se partía.

4.2 Trabajo futuro

Teniendo en cuenta los resultados obtenidos en la realización de este Trabajo Fin de Grado, se plantean los siguientes objetivos como trabajo futuro:

- Combinar ambas partes del TFG para mejorar la eficiencia de los algoritmos de reconocimiento implementados sobre el demostrador mediante técnicas de selección de características basadas en el usuario.
- Probar en otras bases de datos.
- Mejorar los diferentes módulos del esquema de reconocimiento facial, como pueden ser la normalización de la pose o el detector facial. También probar diferentes implementaciones creadas y por crear dentro del estado del arte.
- Sabiendo que las técnicas basadas en la selección de características para un mismo usuario consiguen resultados favorables, seguir esta línea de investigación buscando nuevas hipótesis que mejoren los resultados obtenidos.

Referencias

[1] A. K. Jain, K. Nandakumar, and A. Ross. 50 years of biometric research: Accomplishments, challenges, and opportunities. *Pattern Recognition Letters*, 79:80–105, 2016.

[2] W. Y. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: a literature survey," ACM Comput. Surveys, vol. 35, no. 4, pp. 399–458, 2003.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Proc. Advances Neural Information Processing Systems Conf., 2012, pp. 1097–1105.

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," arXiv Preprint, arXiv:1409.4842, 2014.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Computer Vision Pattern Recognition, 2016, pp. 770–778.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proc. IEEE Conf. Computer Vision Pattern Recognition, 2014, pp. 580–587.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in Proc. Advances Neural Information Processing Systems Conf., 2015, pp. 91–99.

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in Proc. European Conf. Computer Vision, 2016, pp. 21–37.

[9] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa, "An all-inone convolutional neural network for face analysis," in Proc. IEEE Int. Conf. Automatic Face Gesture Recognition, 2017, pp. 17–24

[10] R. Ranjan, V. M. Patel, and R. Chellappa, "A deep pyramid deformable part model for face detection," in Proc. IEEE 7th Int. Conf. Biometrics Theory, Applications and Systems, 2015, pp. 1–8.

[11] R. Ranjan, V. Patel, and R. Chellappa, "Hyperface: a deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," arXiv Preprint, arXiv:1603.01249, 2016.

[12] J. Chen, R. Ranjan, S. Sankaranarayanan, A. Kumar, C. Chen, V. M. Patel, C. D. Castillo, and R. Chellappa, "Unconstrained still/video-based face verification with deep convolutional neural networks," Int. J. Comput. Vis., pp. 1–20. 2017.

[13] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," arXiv Preprint, arXiv:1411.7923, 2014.

[14] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," arXiv Preprint, arXiv:1411.7923, 2014.

[15] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard, "The megaface benchmark: 1 million faces for recognition at scale," in Proc. IEEE Conf. Computer Vision Pattern Recognition, 2016, pp. 4873–4882.

[16] A. Nech and I. Kemelmacher-Shlizerman, “Level playing field for million scale face recognition,” in Proc. IEEE Int. Conf. Computer Vision Pattern Recognition, 2017, pp. 873–4882.

[17] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, “Ms-celeb-1m: A data set and benchmark for large-scale face recognition,” in Proc. European Conf. Computer Vision, 2016, pp. 87–102.

[18] O. M. Parkhi, A. Vedaldi, and A. Zisserman, “Deep face recognition,” in Proc. British Machine Vision Conf., vol. 1, no. 3, 2015, p. 6.

[19] S. Yang, P. Luo, C.-C. Loy, and X. Tang, “Wider face: A face detection benchmark,” in Proc. IEEE Conf. Computer Vision Pattern Recognition, 2016, pp. 5525–5533.

[20] K. Zhang, Z. Zhang, Z. Li, Y. Qiao, "*Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks*", IEEE Signal Processing Letters (SPL), vol. 23, no. 10, pp. 1499-1503, 2016.

[21] J. Neves, J.C. Moreno, H. Proenca, “*Quis-campi: an annotated multi-biometrics data feed from surveillance scenarios*”, IET Biometrics, pp. 1-20, 2018 Neves

[22] E. Gonzçalez Sosa. “*Face and Body Biometrics in the Wild: Advances in the Visible Spectrum and Beyond*”, p. 7, 2017

[23] Disponible en: https://github.com/rcmalli/keras-vggface

[24] R. Ranjan, S. Sankaranarayanan, A. Bansal, N. Bodla, J. C. Chen, V. M. Patel, C. D. Castillo, R. Chellappa, “*Deep Learning for Understanding Faces*”, IEEE Signal Processing Magazine, p. 71, 2018.